ULTRIX

Guide to Software Licensing

Order Number: AA-PBKRA-TE
May 1990

Product Version:	License Management Facility, Version 1.0
Operating System and Version:	ULTRIX, Version 4.0

digital equipment corporation
maynard, massachusetts

Contents

About This Manual

This guide describes the software licensing and license management changes associated with the Digital Distributed Software Licensing Architecture (DDSLA). This includes the new style software license, the Product Authorization Key (PAK) and the License Management Facility (LMF).

The LMF is a software tool for ULTRIX system managers which performs two main functions: license checking and license management.

Audience

This manual is intended for those responsible for managing software licenses on ULTRIX systems. Primarily, this manual is intended for ULTRIX system managers.

This manual also provides information for anyone involved in the use of licensed software on ULTRIX systems.

Organization

This guide is divided into three chapters, an appendix, and a glossary:

Chapter 1	Overview of License Management Provides an introduction to the tools and components associated with the Digital Distributed Software Licensing Architecture (DDSLA)
Chapter 2	PAKs and License Enforcement Describes Product Authorization Keys (PAKs) and how the data they provide is used to prevent unlicensed software use.
Chapter 3	License Management Activities Describes the tasks you can perform using the `lmf`(8) license management utility.
Appendix A	Error Messages Lists and explains the error messages that you may see when you are using the LMF.

Conventions

The following conventions are used in this manual:

bold	Literals are printed in bold type. Literals define specific commands and options which should be entered exactly as shown.
`command(x)`	In text, some mentions of the ULTRIX commands include the section number in the reference manual where the commands are documented. For example, `lmf`(8) appears in Section 8 of the *ULTRIX Reference Pages*.
`example`	In examples, output text is printed in this typeface.
`%`	This is the default user prompt in multiuser mode.
#	This is the default superuser prompt.
italics	In syntax descriptions, this typeface indicates terms that are variable.
`special`	In text, each mention of a specific command, option, partition, path name, directory or file is presented in this typeface.
UPPERCASE	ULTRIX systems differentiate between uppercase and lowercase characters. In examples, enter uppercase characters only where specifically indicated by an example or a syntax line.
[ ]	In syntax descriptions, square brackets indicate terms that are optional.
. . .	In syntax descriptions, a horizontal ellipsis indicates that the preceding item may be repeated.
<KEYNAME>	In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.

The License Management Facility and License Agreements

The terms and conditions of your license agreement determine your legal use of software.

The License Management Facility is a system management tool that can help you comply with your license agreement, but use of the LMF does not indemnify you against non-compliance with the terms and conditions of your software license agreements. In other words, the LMF offers options for many kinds of license agreements, but using some of these options may not be authorized by your specific license agreement.

You must read the terms and conditions of your license carefully to determine which LMF options you can use legally.

This document describes some features of the LMF that Digital is not currently using. Digital may in future use some of the features described herein, but makes no commitment beyond the current Software Business Practices.

Overview of License Management 1

The ULTRIX License Management Facility (LMF) provides software tools for ULTRIX system managers to manage software licenses. The License Management Facility also performs software license checking when software is used that provides full LMF support.

The License Management Facility is part of an approach to software licensing called the Digital Distributed Software Licensing Architecture (DDSLA). DDSLA is an engineering architecture based on the idea of using software license units as a way to size customer computing environments. The license unit is a basic measurement that Digital Equipment Corporation uses to specify how much product use a software license provides. In general, processors that provide more performance need software licenses with more license units.

1.1 The Purpose of DDSLA

When every computer system was a single processor, and software products were dedicated to that processor, both software licensing and the management of software licenses were relatively simple and straightforward. But in today's distributed computing environment, a system manager faces more complexity.

Distributed computing allows a much wider variation in software use than a single processor. Software may be used system-wide, or by just a few users. Who uses it, and where it can be used, may change with the computer environment.

These factors have resulted in a new approach to software licensing and the introduction of new tools for managing license data.

The Digital Distributed Software Licensing Architecture (DDSLA) is an approach to software licensing that:

- Is consistent for all ULTRIX products
- Provides for organized record-keeping
- Includes useful tools to help you monitor and control software use

1.2 The Components of DDSLA

The implementation of the Digital Distributed Software Licensing Architecture (DDSLA) consists of:

- The License Management Facility
- Product Authorization Keys (PAKs)

The License Management Facility is part of the ULTRIX operating system and consists of:

- A License Database (LDB)
- License Unit Requirement Tables (LURTs)
- A license management utility, `lmf`(8)
- License-checking functions which are included in the licensed products

A Product Authorization Key (PAK) is a set of license information which the License Management Facility uses to confirm that a product is licensed. Licenses are normally printed on paper and sent to you as part of the product kit.

When you receive a PAK you should enter the license information into the License Database using the `lmf`(8) utility. This is called ''registering a license'' and some licensed products will not run unless you have registered a license for them. In some cases you may need to register a license before you install the product. The License Database is automatically created when you register the first license. The PAK is your proof of license, and should be stored in your files for future reference.

While the number of license units on the PAK defines the size of the license, each processor has a series of license unit requirements, also specified in license units. License Unit Requirement Tables define the number of license units required to run a product on a particular size of processor. The LMF compares the size of a registered license to the license unit requirement for the processor, and authorizes product use when a license supplies sufficient license units.

1.3 Types of Software License

There are two basic types of license:

- Availability (or Capacity) Licenses
- Activity (or Per-user) Licenses

The next two sections describe each type.

1.3.1 Availability Licenses

Availability Licenses are also called Capacity Licenses. Availability Licenses only allow a product to run on certain types of executing hardware; the more powerful the hardware, the more license units are needed for each product. Once a product has been licensed on a particular processor, users have unlimited access to the product. Unlike previous Capacity Licenses, DDSLA defined Availability Licenses are not restricted to a particular processor before they are registered; that is, the license does not state which processor it must be registered on.

1.3.2 Activity Licenses

Activity Licenses are also called Per-user Licenses. Activity Licenses restrict the number of simultaneous users of a product. The number of licensed users can be increased by purchasing additional license units.

1.4 Your Responsibility in License Management

Software is provided to customers under an agreement called a license. A software license can involve a rental agreement and other complex arrangements. Although the term *license* can have specific legal meanings, for the purposes of this manual a

license refers to the authorization you have to use a product.

Note

Digital provides the License Management Facility to help you with the task of license management. However, the responsibility remains with your company for using the tools in a way that fulfills your record-keeping obligations, and for ensuring that your company honors all license terms.

PAKs and License Enforcement 2

Product Authorization Keys (PAKs) and license checking by the License Management Facility are being gradually introduced for ULTRIX software. This chapter is designed to help you understand PAKs and how license enforcement by the LMF will affect your use of ULTRIX software.

This chapter describes:

- Which products are affected by the LMF
- Product Authorization Keys
- License enforcement for availability licensed and activity licensed products

2.1 Products Affected by the LMF

In general, Digital Equipment Corporation's software products fall into two categories:

- Software unaffected by the LMF
  These products do not use the LMF to authorize software access. License information is provided separate from the software and may be in the form of a PAK which can be registered in the License Database.
- Software that provides full LMF support
  License information is provided by a PAK which must be registered in the LDB. The LMF checks for unauthorized use of these products.

The release date of a product, relative to the release date of ULTRIX V4.0, normally determines how much support a product provides for the LMF:

- Software released before ULTRIX V4.0 is unaffected by the LMF.
- Software released after ULTRIX V4.0 provides full LMF support.

The software that could be affected by the LMF includes Digital layered products and operating systems based on ULTRIX, such as ULTRIX Worksystems Software (UWS).

Note

Layered products that provide full LMF support can only be used on operating systems based on ULTRIX V4.0 (or later). Operating systems based on prior releases of ULTRIX do not have the License Management Facility to authorize access to these products.

The LMF is not designed to be used exclusively by Digital products. Other companies may issue PAKs, or have Digital issue them on their behalf, and include license-checking functions in their software. However, for clarity, in this manual it is assumed that all software is supplied and produced by Digital.

You should refer to the product documentation to find out if the product provides support for the License Management Facility.

2.2 Product Authorization Keys

A Product Authorization Key is a unique set of data generated by Digital and used by the LMF to confirm that a product is licensed.

The License PAK is the ''standard'' PAK, provided when you purchase a software license. It is a valuable proof of purchase, represents your license from Digital to use a software product, and should be stored in your files. The license information is confidential and should not be publicly posted or widely distributed. In order to comply with Digital's license terms, you must always register a License PAK in the License Database.

2.2.1 Obtaining a PAK

Generally, you obtain both a PAK and the product from the Digital representative who distributes software. You order a PAK just as you might order another product from Digital. Before you order a PAK, you should define your software and hardware requirements to your Digital representative so that you get a license of the correct size. You will normally receive a PAK printed on a piece of paper when you buy the software.

2.3 Other Types of PAK

As well as License Product Authorization Keys, there are three other types of PAKs Digital uses to cover various situations:

- Service Update PAK
- Temporary Service PAK
- Product Authorization Amendment

2.3.1 Service Update PAK (SUP)

This is a transitional tool provided to software maintenance customers as part of a layered product's next scheduled update, when the product provides technical support for the LMF. A Service Update PAK contains the same fields as a License PAK and the information should be registered in the LDB using the `lmf register` command. For more information on `lmf register`, see Section 3.2. Unlike a License PAK, a SUP does not represent a software license; it is merely a temporary measure that you use if you do not yet have a License PAK for a layered product. A License PAK always supersedes a Service Update PAK.

2.3.2 Temporary Service PAK (TSP)

This is a temporary exception measure used by Digital Field Service in performing customer support activities. A Temporary Service PAK contains the same fields as a License PAK and the information should be registered in the LDB using the `lmf register` command (see Section 3.2). Again, this type of PAK does not represent a license and will be replaced with a License PAK if appropriate.

2.3.3 Product Authorization Amendment

A Product Authorization Amendment (PAAM) is similar to a License PAK but only includes the data needed to identify, update, and further authorize product use. For example, you may receive a PAAM if you want to increase the number of units for a license that does not have the MOD_UNITS option. The license for the product should be updated using the `lmf amend` command. For more information on `lmf amend`, see Section 3.12.2.

2.4 Information on a PAK

This section describes the information contained in the fields on the PAK. A typical PAK is shown in Figure 2.1 When you register the PAK in the LDB, you will be provided with a Comments field that you can use as needed. Section 3.2 describes how to register a PAK.

Figure 2-1: A Typical PAK

```
 - - - - - - -
| | | | | | | |                    LICENSE PAK
|d|i|g|i|t|a|l|             (PRODUCT AUTHORIZATION KEY)
| | | | | | | |
 - - - - - - -
Digital Equipment Corporation


The Software License Product Authorization Key is provided subject to
terms appearing on the back of this document.

                    Product: ALLSUM ACCOUNTING TOOL

License Model No. QL-010AB-AA    DEC No. 125436    Issue Date: 1-JAN-1989

*************************************************************************

                     ISSUER:          DEC

      AUTHORIZATION NUMBER:          AWS-PK-88229-2

               PRODUCT NAME:         ALLSUM

                  PRODUCER:          DEC

             NUMBER OF UNITS:        1200

                    VERSION:

      PRODUCT RELEASE DATE:          1-JUL-1991

      KEY TERMINATION DATE:

    AVAILABILITY TABLE CODE:         M

       ACTIVITY TABLE CODE:

                KEY OPTIONS:

              PRODUCT TOKEN:

               HARDWARE I.D:

                  CHECKSUM:          1-ODKM-NIIO-JEPJ-FCLB

*************************************************************************
```

2.4.1 Issuer

The Issuer is the LMF name for the entity that supplies the PAK. Most licenses specify ''DEC'' for the PAK issuer. However, Digital may further identify PAK issuers by region or department within the company. For example, the PAK issuer string could be ''DEC-USA'' or ''DEC-EUROPE.'' Other software vendors with products that provide support for the LMF can also issue PAKs.

2.4.2 Authorization Number

The Authorization Number, together with the Issuer, uniquely identifies each license, both for you and for Digital. It helps tasks such as reconciling records between customers and Digital, because it allows everyone to know immediately and with certainty which license is being referred to.

2.4.3 Product Name

The Product Name is the name used by the LMF to distinguish between different products. The Product Name that appears on the PAK may be slightly different from that in the Software Product Description. This is due to restrictions imposed by the LMF.

2.4.4 Producer

The Producer is the name of the company producing the software. For all software produced by Digital, the producer name will be ''DEC.'' The Producer is used by the LMF to distinguish between products with the same name but produced by a different company. For example, you might have two FORTRAN compilers, one produced by Digital and one produced by another company.

2.4.5 Number of Units

The Number of Units shows how many license units have been supplied with the PAK. Licenses with a Number of Units shown as zero (0) indicate that the license has unlimited size, that is, it provides unlimited use of the product on any type of processor.

2.4.6 Version

This is the software product version number. The Version will not appear on every product's PAK. It is used by the LMF to restrict the use of the PAK to particular versions of the product. For example, if the Version number on the PAK is 2.0, then the PAK may be used with all versions of the product up to and including Version 2.0.

2.4.7 Product Release Date

The Product Release Date will not appear on every product's PAK. The Product Release Date is used by the LMF to restrict the use of the PAK to versions of the product released before a certain date. For example, if the Product Release Date on the PAK is 1-JUL-1991, then the PAK is valid for use with all versions of the product released on or before that date.

Digital does not issue PAKs that have both the Version and the Product Release Date on them. Some PAKs may be issued that are not restricted by Version or by Product Release Date.

2.4.8 Key Termination Date

This is the termination date of the PAK. After this date, the PAK no longer represents a valid license for the product.

2.4.9 Availability Table Code

This represents the number of units required to give unlimited use on a particular processor. If the product is availability licensed, this field will contain a letter or "CONSTANT=*integer.*" A letter represents the License Unit Requirement Table that defines the number of units required for the product to run on a particular processor. If your PAK has an Availability Table Code with, for example, "CONSTANT=100," it means that the product needs 100 units to run on any type of processor, regardless of size. For a complete explanation of how the Availability Table Code is used, see Section 2.6.1.

2.4.10 Activity Table Code

This represents the number of units required for each simultaneous user of the product. If the product is activity licensed, this field will contain a letter or "CONSTANT=*integer.*" A letter represents the License Unit Requirement Table that defines the number of units required for each simultaneous user to run the product on a particular processor. If your PAK has an Activity Table Code with, for example, "CONSTANT=100," it means that each simultaneous user of the product needs 100 units to run the product on any type of processor, regardless of size. For a complete explanation of how the Activity Table Code is used, see Section 2.6.2.

2.4.11 Key Options

There are three options which can appear in this field:

- MOD_UNITS
- NO_SHARE
- P_FAMILY

The MOD_UNITS option indicates that you may modify the Number of Units field (using `lmf modify`). For a complete description of how to modify the Number of Units, see Section 3.12.1.

The NO_SHARE option indicates that you cannot combine two or more licenses for the product on the same processor. For a complete description of license combination, see Section 3.20.

The P_FAMILY option indicates that the LMF will attempt to allocate license units for the product to only one executing process, even if a user has several processes executing the same licensed software.

2.4.12 Product Token

This field is not currently used by the LMF. However, any data that is in this field must still be entered into the License Database to prevent a Checksum error.

2.4.13 Hardware-Id

This field is not currently used by the LMF. However, any data that is in this field must still be entered into the LDB to prevent a Checksum error.

2.4.14 Checksum

The Checksum is generated from the individual data elements on the PAK. The Checksum will be unique for each PAK and ensures that you have entered the PAK data correctly into the LDB.

2.5 License Unit Requirement Tables

License Unit Requirement Tables (LURTs) are provided as part of the License Management Facility. LURTs are a series of tables that specify a series of license units requirements, essentially performance ratings, for each System Marketing Model (SMM). Although this manual generally refers to computer systems as processors, the LMF actually identifies a system by its System Marketing Model, which is the model name used in marketing and pricing. The SMM generally corresponds to the name on the front panel of the processor cabinet.

Each LURT has a rating, in license units, for all currently available (and appropriate) processors. For example, the ULTRIX Layered Product LURT includes every processor that can run ULTRIX, and associates a number of units with each. When Digital releases new processors, the tables are updated as part of the new processor support. There are 14 LURTs, with the following codes and types:

A – H	VMS Tables
J	ULTRIX Capacity
K	ULTRIX 'n' User
L	System Integrated Products
M	Standard Layered Products
N	Reserved for future use
P	Reserved for future use

The LURT tables are not stored in a readable form on ULTRIX systems, but this will not prevent you ordering licenses of an appropriate size for your system. When you order a product, you should define your software and hardware needs to your Digital representative, who will ensure you receive a license with the appropriate number of units.

2.6 License Checking

The LMF authorizes a product to run only if there is a valid license for the product. Products that provide technical support for the LMF in their software have license-checking functions that check the following:

- Software has the same Product Name and Producer name as those on the license
- Version number of the software is not greater than the Version number (if specified) on the license
- Product release date of the software is not later than the Product Release Date (if specified) on the license
- Current date is not later than the Key Termination Date (if specified) on the license
- Current date is not later than the Cancellation Date (if specified) on the license

The license-checking functions check the license details in the kernel cache. The kernel cache contains details of those licenses that have been registered in the License Database and subsequently loaded into the kernel cache. License details are loaded into the kernel cache when the following occurs:

- The system is rebooted
- The `lmf load` command is used (see Section 3.5.1)
- The `lmf reset` command is used (see Section 3.5.2)

Note

You must load a valid license for a product into the kernel cache before attempting to access the product.

The number of license units registered with any license should match or exceed the number of license units required for the specified product to run on the specified processor. For example, suppose you obtain a license for the fictional product ALLSUM to run on a MicroVAX II. That ALLSUM license should specify at least the same number of license units as the ULTRIX Layered Product LURT requires for a MicroVAX II. The same license may not provide enough license units to authorize use of the product on a VAX 8800.

For a complete explanation of license checking for availability licensed products, see Section 2.6.1. For a complete explanation of license checking for activity licensed products, see Section 2.6.2.

2.6.1 Availability Licensed Products

A valid Availability License makes a product available to all the users of a system. The LMF makes a product accessible if the number of units on the license matches or exceeds the license unit requirement for the current processor. Availability licenses are checked:

- By the LMF, when the license details are loaded into the kernel cache from the LDB
- By the license-checking function in the product, when a user attempts to access the licensed software

2.6.1.1 Loading a License into the Kernel Cache – When you load a license into the kernel cache the LMF looks for the Availability Table Code field of the registered license. If the license specifies ''CONSTANT=*integer*,'' the LMF defines the license unit requirement as equal to the stated integer value. This value can be the decimal value zero (0), which means the license has no unit requirements.

If the license does not specify a constant unit requirement, the LMF looks for a code that corresponds to an entry in the LURT. The LMF determines the System Marketing Model of the current processor and locates the SMM in the appropriate LURT. The LMF selects the value that specifies the number of units required and compares this against the number of units on the license.

If the number of units on the license matches or exceeds the license unit requirement for the current processor, license details from the License Database are copied into the kernel cache, and the product becomes accessible by all users on the system.

If the number of units on the license is less than the license unit requirement for the current processor, the license details are not copied into the kernel cache and an error message is displayed. If you are trying to load a license into the kernel cache with the `lmf reset`, command, or by rebooting the system, the error message displayed is:

`Not enough units to load` *product producer*

If you are trying to load a license into the kernel cache with the `lmf load` command, the error message displayed is:

```
License too small to load this many users
```

Consider an example where all the PAKs for the fictional layered software product ALLSUM refer to LURT M, designated by the letter M next to the Availability Table Code field on each PAK. Your PAK for ALLSUM may also provide 1000 license units, designated by the number 1000 next to the number of units field on the PAK.

When you register and load the 1000-unit license, the LMF selects LURT M, and compares the license unit value 1000 to the value found in LURT M next to the current SMM. For this example, assume the current processor, VAXMID, requires 1000 license units to activate a layered product in LURT M. The LMF allows the license details from the License Database to be copied into the kernel cache and the product becomes accessible by all users on the system.

Now consider the current processor to be VAXBIG, and assume VAXBIG requires a 1500-unit license to activate the product ALLSUM. Because the number of units on the license is less than the license unit requirement for the current processor, the license details are not copied into the kernel cache, and the appropriate error message is displayed.

2.6.1.2 Accessing the Licensed Software – Each time a user attempts to access an availability licensed product, the license-checking function in the layered product checks the kernel cache for a valid license for the product; that is, it performs the checks described in Section 2.6. No check is made on the number of license units for the product because the LMF only allows availability licenses with a sufficient number of units to be loaded into the kernel cache.

Availability licensed products that have a valid license in the kernel cache can be accessed by all users on the system.

If a user attempts to access an availability licensed product that does not have a valid license in the kernel cache, the license-checking function prevents access to the

product and displays the following message on the terminal:

```
No license found for this product
```

This situation occurs when, for example, you have registered the license details for the product in the LDB and installed the software, but you have not yet rebooted the system or used `lmf load` or `lmf reset`.

Note

Not all license-checking functions behave as described in the previous section. Some may prevent access to the product without displaying an error message, others may allow users to access the product, even if the product does not have a valid license in the kernel cache. You should refer to the documentation supplied with the layered product to find out exactly what action the checking function will take.

2.6.2 Activity Licensed Products

An Activity License defines the number of simultaneous users allowed for a product. The LMF makes a product available to a user if the number of units on the license matches or exceeds the license unit requirement for the current processor. Activity license are checked:

- By the LMF, when the license details are loaded into the kernel cache from the LDB
- By the license-checking function in the product, when a user attempts to access the licensed software

2.6.2.1 Loading a License into the Kernel Cache – When you load a license into the kernel cache the LMF looks for the Activity Table Code field of the registered license. If the license specifies ''CONSTANT=*integer*,'' the LMF defines the license unit requirement for each user as equal to the stated integer value. This value can be the decimal value zero (0), which means each user has no unit requirements.

If the license does not specify a constant unit requirement, the LMF looks for a code that corresponds to an entry in the LURT. The LMF determines the System Marketing Model of the current processor, locates the SMM in the appropriate LURT and selects the value that specifies the number of units per user required.

The number of units per user required by the current processor and the license details from the LDB are copied into the kernel cache.

2.6.2.2 Accessing the Licensed Software – Each time a user attempts to access an activity licensed product, the license-checking function in the product checks the kernel cache for a valid license for the product, that is, it performs the checks described in Section 2.6.

If the product has a valid license in the kernel cache, the license-checking function in the product compares the number of units required for each user to the number of units available. If the number of units available matches or exceeds the license unit requirement for the current processor, the user can access the product. When the license-checking function allows the first user to access a product, it allocates the number of units required from the kernel cache. Using the fictional ALLSUM again, the license-checking function may allocate 25 of a registered 100 units to the first

user. As long as the first user is using the product, those 25 units remain allocated, leaving 75 available in the kernel cache for other users. If the number of units on the license is less than the license unit requirement for the current processor, the user is refused access to the product and the following message is displayed at the terminal:

```
Attempted usage exceeds active license units
```

When the next user attempts to use the product, the checking function repeats the authorization procedure again. For example, when the second user invokes ALLSUM, the checking function looks for 25 available license units to authorize product use. Because the ALLSUM license now has 75 license units unallocated in the kernel cache, the license-checking function again authorizes product use. In this example, the first four concurrent users can access the product, but additional users are denied access.

As each user finishes using the product, the kernel returns the allocated units for another user.

If a user attempts to access an activity licensed product that does not have a valid license in the kernel cache, the license checking function prevents access to the product and displays the following message on the terminal:

```
No license found for this product
```

This situation occurs when, for example, you have registered the license details for the product in the LDB and installed the software, but you have not yet rebooted the system or used `lmf load` or `lmf reset`.

Note

Not all license-checking functions behave as described in the previous section. Some may prevent access to the product without displaying an error message, others may allow users to access the product, even if the product does not have a valid license in the kernel cache. You should refer to the documentation supplied with the layered product to find out exactly what action the checking function will take.

License Management Activities 3

The License Management Facility provides the `lmf` utility to help manage the software licenses for your system. The `lmf` utility maintains a file of registered software licenses, the License Database. In addition to maintaining the LDB, you can also use the `lmf` utility to control the access to licensed software on the system. This chapter describes how to:

- Use the `lmf` utility
- Register a license
- Activate a license
- Update a license
- Restrict the use of a product
- Disable a license
- Cancel a license
- Delete a license
- Monitor the LDB and kernel cache
- Review your license management activities
- Change the number of active CPUs
- Combine licenses
- Manage availability licensed products
- Manage activity licensed products

3.1 Using the lmf Utility

The `lmf` commands can only be used by a person logged into the system as the superuser (root login). You can allow nonprivileged users to use the `lmf list` and `lmf history` commands, but you need to change the file mode permissions on the files the commands access. The `lmf list` command accesses `/usr/var/adm/lmf/ldb`, and the `lmf history` command accesses `/usr/var/adm/lmf/ldb_history`. You can change the directory containing the LDB file and the history file by using the `-d` *dir* option. This allows you to have more than one LDB on your system.

When you use the `lmf` commands you can type them on a single line, for example:

```
# lmf register
```

You can also enter the `lmf` utility and type the commands after the prompt, for example:

```
# lmf
```

```
lmf> register
```

3.2 Registering a License

This section describes how to use the `lmf register` command to register license details from a Product Authorization Key in the License Database. Using `lmf register` you can:

- Edit an empty template and register the license details from the completed template
- Edit an existing template and register the license details from the completed template
- Register details directly from a file or electronic-mail message

The LMF checks the license details you are trying to register, to ensure that there are entries against all the appropriate fields. The following fields always require an entry:

- Issuer
- Authorization Number
- Product Name
- Availability Table Code or Activity Table Code, or both
- Checksum

If the Producer field is blank, the LMF assumes the Producer to be "DEC." The LMF ensures there are entries against all the mandatory fields and that the Checksum validates all the license data. Licenses with incorrect or missing entries are not registered in the LDB.

The following sections describe the three ways to register licenses in the LDB and explain what to do if something goes wrong.

3.2.1 Editing an Empty Template

Use the `lmf register` command with no arguments to add license details to an empty template and register the details in the LDB. The command displays a template which includes all the fields on a PAK and an additional field for your comments. An editor is invoked so that you can add the license data to the appropriate fields. The editor used is defined by the EDITOR environment variable. If this is not set, `/usr/ucb/vi` is used.

You need to use `lmf register` to register license information from a paper License PAK, see Figure 3.1.

Figure 3-1: A Typical PAK

```
- - - - - - -
| | | | | | | |                    LICENSE PAK
|d|i|g|i|t|a|l|              (PRODUCT AUTHORIZATION KEY)
| | | | | | | |
 - - - - - - -
Digital Equipment Corporation


The Software License Product Authorization Key is provided subject to
terms appearing on the back of this document.

                  Product: ULTRIX OPERATING SYSTEM

License Model No. QR-234AB-DC   DEC No. 458394    Issue Date: 1-JAN-1989

**************************************************************************

                   ISSUER:          DEC

     AUTHORIZATION NUMBER:          TPQ-PK-88229-23

              PRODUCT NAME:         ULTRIX

                  PRODUCER:         DEC

            NUMBER OF UNITS:        1600

                   VERSION:

     PRODUCT RELEASE DATE:          1-JUL-1991

     KEY TERMINATION DATE:

  AVAILABILITY TABLE CODE:

      ACTIVITY TABLE CODE:          CONSTANT=100

               KEY OPTIONS:         NO_SHARE,P_FAMILY

             PRODUCT TOKEN:

              HARDWARE I.D:

                  CHECKSUM:         1-AFAP-ICFD-BCPJ-FCGL

**************************************************************************
```

To register the license data from the example PAK you type:

```
# lmf register
```

The command displays an empty template, and you should use the editor to type in the PAK data against the appropriate field:

```
                Licensed Software Product
                Product Authorization Key

 Enter data on lines terminated with :

                  Issuer: dec
   Authorization Number: tpq-pk-88229-23

            Product Name: ultrix
                Producer: dec

          Number of units: 1600

                 Version:
   Product Release Date: 1-jul-1991

   Key Termination Date:

Availability Table Code:
    Activity Table Code: constant=100

             Key Options: no_share,p_family
           Product Token:
             Hardware-Id:
                Checksum: 1-afap-icfd-bcpj-fcgl

Comment: This is an example license
```

Any fields that are blank on the PAK should be left blank when you type in the data. You can type the data in uppercase or lowercase; when the data is copied into the LDB, it is automatically put into uppercase.

Note

> You must type in the license information from the PAK carefully. The LMF may return only a checksum error message if you omit or incorrectly enter any license data. Carefully check the characters typed on each line, not just the checksum string.

When you leave the editor, the LMF scans the completed template to ensure that all the license data has been entered correctly. If the license data is correct, it is copied into the License Database. If the license data is incorrect, you are given the opportunity to re-enter the editor and correct any mistakes.

3.2.2 Editing an Existing Template

Use the `lmf register` *filename* command to edit and register license data from a file on your system. The editor invoked is defined by the EDITOR environment variable. If this is not set, `/usr/ucb/vi` is used.

You may have license data in files on your system as a result of using the `lmf issue` command (see Section 3.9), or they may be copied on to your system as part of a product installation. For example, the installation software for ULTRIX operating systems copies license data to the file `/usr/var/adm/lmf/ultrix`. This file contains license data common to all ULTRIX operating system licenses. To register the license, you just need to add your specific license details from your ULTRIX PAK. Using the example PAK shown in Figure 3-1, to register the

additional data on to the file created during installation, type the following:

```
# lmf register /usr/var/adm/lmf/ultrix
```

The command displays the file, in this example `/usr/var/adm/lmf/ultrix`, and you should use the editor to complete the license details:

```
                Licensed Software Product
                Product Authorization Key

 Enter data on lines terminated with :

                Issuer: DEC
   Authorization Number: tpq-pk-88229-23

          Product Name: ULTRIX
              Producer: DEC

        Number of units: 1600

               Version:
   Product Release Date: 1-JUL-1991

   Key Termination Date:

Availability Table Code:
    Activity Table Code: CONSTANT=100

           Key Options: NO_SHARE,P_FAMILY
         Product Token:
           Hardware-Id:
              Checksum: 1-afap-icfd-bcpj-fcgl

Comment: This is an example license
```

You must ensure that the file you are registering contains all the license fields that have entries. The license fields must be in the same format as the template displayed with `lmf register`, that is, the same combination of uppercase and lowercase letters, with a colon (:) separating the field name and the data. The license fields can be in any order. Files with license data created using the `lmf issue` command or as part of product installations automatically have the field names in the correct format.

When you leave the editor, the LMF scans the completed template to ensure that all the license data has been entered correctly. If the license data is correct, it is copied into the License Database. If the license data is incorrect, you are given the opportunity to re-enter the editor and correct any mistakes.

3.2.3 Registering a PAK Directly

Use the `lmf register - <` *filename* command to register a file containing PAK data. The file can be created by the `lmf issue` command or may be an electronic-mail message containing PAK data.

The command does not display the contents of the file or allow you to edit it. However, the LMF does scan the file to ensure the format and data are correct. If the license data is correct, it is copied into the LDB. If the license data is incorrect, it is not copied into the LDB and the appropriate error message is displayed. The command also returns an error status: zero (0) if the license data has been copied into the LDB, or nonzero, if the license data has not been copied into the LDB.

3.3 Registering Licenses for ULTRIX Systems

This section describes how to register the license for your ULTRIX operating system. It provides an overview of the licensing events that occur when you install your ULTRIX system and describes two situations in particular:

- Installing a new ULTRIX system
- Upgrading an existing ULTRIX system

When you install your ULTRIX operating system, the installation software reboots the system. When a system is rebooted, the LMF tries to load the kernel cache with the license details from the License Database. If the LMF finds an ULTRIX license in the LDB file (`/usr/var/adm/lmf/ldb`), it (as well as any other licenses) is loaded into the kernel cache. If the ULTRIX license was generated automatically or if there is no ULTRIX license, the LMF searches for the `/upgrade` file. The `/upgrade` file contains the maximum number of concurrent users allowed for ULTRIX.

If the `/upgrade` file exists, the LMF loads enough license units into the kernel cache to allow the number of concurrent users specified by the `/upgrade` file. The LMF also creates an entry for an ULTRIX license in the LDB, with a comment saying that the entry was created automatically.

If the `/upgrade` file does not exist, the LMF loads enough units into the kernel cache to allow two concurrent users. The LMF does not create an entry in the LDB for ULTRIX, if there is no `/upgrade` file.

When you receive your Product Authorization Key for ULTRIX, you should use the `lmf delete` command to remove the ULTRIX license data generated automatically by the LMF. Register the PAK for ULTRIX using the `lmf register` command and then load the license into the kernel cache using the `lmf load` command. The following sections provide more details on these commands.

3.3.1 New Systems

When you install ULTRIX on a new system and the installation software reboots the system, the LMF loads enough units into the kernel cache to allow two concurrent users.

When you receive your ULTRIX PAK, you should register it in the LDB. License data common to all ULTRIX operating system licenses is contained in the file `/usr/var/adm/lmf/ultrix`. You can use this data when you register your ULTRIX PAK. To display `/usr/var/adm/lmf/ultrix`, type the following command line:

```
# lmf register /usr/var/adm/lmf/ultrix
```

Use the editor and the information on your ULTRIX PAK to complete the license details (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the command:

```
# lmf load 0 ultrix
```

3.3.2 Existing Systems

When you install ULTRIX on an existing system there may be an existing LDB or a binary upgrade file. If there is an ULTRIX license in the LDB, when the installation software reboots the system the LMF loads the license into the kernel cache. In this case you need take no further action to register your ULTRIX license.

If the ULTRIX license was generated automatically or if there is no ULTRIX license when the system reboots, the LMF searches for the `/upgrade` file. If the `/upgrade` file exists, the LMF loads enough license units into the kernel cache to allow the number of concurrent users specified by the `/upgrade` file. The LMF also creates an entry for an ULTRIX license in the LDB, with a comment saying that the entry was created automatically.

If the `/upgrade` file does not exist, the LMF loads enough units into the kernel cache to allow two concurrent users. The LMF does not create an entry in the LDB for ULTRIX, if there is no `/upgrade` file.

When you receive your ULTRIX PAK, you should remove the ULTRIX license generated automatically by the LMF. Use the command:

```
# lmf unload 0 ultrix
```

This unloads the ULTRIX license from the kernel cache. If the ULTRIX license was created without using data from the `/upgrade` file, there is no entry in the LDB for you to delete. If the ULTRIX license was created from data in the `/upgrade` file, use the command:

```
# lmf delete ultrix
```

This deletes the license from the LDB.

The installation software copies license data common to all ULTRIX operating systems licenses to the file, `/usr/var/adm/lmf/ultrix`. You can use this data when you register your ULTRIX PAK. Use the command:

```
# lmf register /usr/var/adm/lmf/ultrix
```

This displays `/usr/var/adm/lmf/ultrix`. Use the editor and the information on your ULTRIX PAK to complete the license details (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the command:

```
# lmf load 0 ultrix
```

3.4 Registering Licenses for Layered Products

You can register a license for a layered product by:

- Editing a blank template
- Editing an existing file

3.4.1 Editing an Empty Template

Use this method to register a license for a product that:

- Has not been installed yet
- Does not create a file containing license data as part of its installation procedure

Some layered products require a valid license in the kernel cache before they can be installed. These layered products run as part of their installation procedure.

To edit an empty template, use the command:

```
# lmf register
```

This displays a template which includes all the fields on a PAK. You should use the editor to type in the PAK data against the appropriate fields on the template (see Section 3.2.1)

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the following syntax:

`lmf load 0` *product_name*

In this command example, *product_name* is the same as the Product Name on the PAK.

3.4.2 Editing an Existing Template

Some layered products create a file containing license data as part of their installation procedure. The license data is common to all PAKs for the product; to register the license, you just need to add your specific license details from the PAK you receive. The license data is copied to the file `/usr/var/adm/lmf/`*product_name* where *product_name* is the LMF Product Name as it appears on the PAK. For example, the product ALLSUM would copy license data to the file `/usr/var/adm/lmf/allsum`.

To register a license using data from a file created as part of the product installation, use the command:

`lmf register /usr/var/adm/lmf/`*product_name*

Use the editor to type in the PAK data against the appropriate fields on the template (see Section 3.2.2).

When you have left the editor and the license has been registered in the LDB, you should load the license into the kernel cache using the following command syntax:

`lmf load 0` *product_name*

3.5 Loading a License into the Kernel Cache

When you have registered a license in the License Database, you should load it into the kernel cache to make the license details available to the license checking functions. The license checking functions allow a product to run only if it has a valid license in the kernel cache.

To load license details into the kernel cache, copy license details from the LDB by:

- Using the `lmf load` command. This command copies the license details for a particular product from the LDB to the kernel cache.
- Using the `lmf reset` command. This command copies the license details for all products from the LDB to the kernel cache.
- Rebooting the system. The reboot process automatically executes the `lmf reset cpus` command.

3.5.1 Loading a License for One Product

Use the `lmf load` command to copy the license details for a specified product from the LDB to the kernel cache. The `lmf load` command has the following syntax (see also the `lmf`(8) reference page):

`lmf load` *users product* [*producer* [*authorization*]]

The LMF loads the number of license units corresponding to the number of *users* specified in the command into the kernel cache (assuming that there are enough license units registered in the LDB). If you specify zero (0) as the *users* argument, the LMF loads all the license units registered for the product in the LDB into the kernel cache.

If the product is availability licensed, you must specify the number of users to be zero when you use the `lmf load` command. This ensures that the number of units loaded into the kernel cache is always enough to satisfy the requirements of the processor. For example, assume the product ALLSUM is availability licensed, and you have registered the license in the LDB. To load the license into the kernel cache, you should type:

```
# lmf load 0 ALLSUM
```

Alternatively, assume the product ALLSUM is activity licensed, and you have registered a 10-user license for the product in the LDB. To load all the license units for the product into the kernel cache, you should type:

```
# lmf load 0 ALLSUM
```

If you only wanted to load enough license units for 5 users, you should type:

```
# lmf load 5 ALLSUM
```

When you use the `lmf load` command you must ensure that you supply enough arguments to uniquely identify the license. If you have the same product but from different producers, you must supply the producer name as well as the product name, for example:

```
# lmf load 0 ALLSUM DEC
```

If there are two or more licenses with the same product and producer name, the `load` command loads all the licenses into the kernel cache only if the licenses can be combined. For a complete explanation of license combination, see Section 3.20.

3.5.2 Loading the Licenses for All Products

Use the `lmf reset` command to copy the license details for all products from the LDB to the kernel cache. The `lmf reset` command has the following syntax (see also the `lmf`(8) reference page).

`lmf reset [ cpus [ n ]]`

In addition to copying the license details from the LDB to the kernel cache, `lmf reset cpus` checks the number of active CPUs and uses this number to determine the System Marketing Model. The SMM is used by some products to define the number of license units needed in the kernel cache before access to the product is granted.

The *n* argument represents the number of active CPUs on the system when determining the SMM.

3.6 Unloading a License from the Kernel Cache

Use the `lmf unload` command to restrict the number of users of a product. You can do this by removing license units from the kernel cache, thus restricting the number of units available to the LMF checking functions. The command affects only the number of license units available for a product in the kernel cache; it does not affect the number shown in the LDB.

For example, suppose you have registered and loaded the license for the product ALLSUM with a 10-user Activity License. You could restrict the license to a 7-user license by typing:

```
# lmf unload 3 ALLSUM
```

Existing users of the product are allowed to finish using it before the new limit is imposed. For example, if there are 10 users of a product and the `lmf unload` command is used to restrict the number of users to 7, all 10 users will be able to finish using the product. However, new users of the product will not be allowed until the number of current users has dropped to less than the new limit of 7.

In the case of an availability licensed product, you must unload all the license units for the product. You do this by specifying zero (0) as the number of users. This indicates that all the license units for the product should be removed from the kernel cache. For example, if the product ALLSUM was availability licensed, and you wanted to unload the license units for the product, you would type:

```
# lmf unload 0 ALLSUM
```

As with activity licensed products, existing users of the product are allowed to finish using it, but new users are refused access.

If you do not want the license to be reloaded when the system is rebooted or when you issue the `lmf reset` command, you should disable the license with the `lmf disable` command.

3.7 Enabling a License

Use the `lmf enable` command to enable a license to be loaded into the kernel cache. If a license is disabled, it is ignored when you use `lmf load`, `lmf reset` or when you reboot the system.

For example, to enable the license for the product ALLSUM and load the license into the kernel cache, you type:

```
# lmf enable ALLSUM
# lmf load 0 ALLSUM
```

When you register a license in the LDB, it is automatically enabled; that is, you can load it into the kernel cache immediately.

3.8 Disabling a License

Use the `lmf disable` command to prevent a license from being loaded into the kernel cache when you use the `lmf load` command, the `lmf reset` command, or reboot the system. For example, to disable the license for the product ALLSUM, you would type:

```
# lmf disable ALLSUM
# lmf unload 0 ALLSUM
```

The `lmf disable` command does not immediately affect the kernel cache, so you should use the `lmf unload` command to unload the license details from the kernel cache. If you do not use the `lmf unload` command, the product remains available on the system until the next `lmf reset` command or system reboot. You cannot disable the ULTRIX license generated by the LMF from the `/upgrade` file.

3.9 Issuing a License

Use the `lmf issue` command to move the license details for a product from the LDB to a file on your system. The command reconstructs a Product Authorization Key from the license data in the LDB and outputs the PAK to a specified file. If the PAK is issued correctly, the license is deleted from the LDB and unloaded from the kernel cache. You can use this command to move a license for a product from one system to another (see Sections 3.16 and 3.18).

The `lmf issue` command has the following syntax (see also the `lmf`(8) reference page):

`lmf issue` *file product* [*producer* [*authorization*]]

For example, to issue the license for the product ALLSUM to the file `allsum.pak` type:

```
# lmf issue allsum.pak ALLSUM
```

Although the command removes the license data from the kernel cache, existing users of the product are allowed to finish using it. You cannot issue the ULTRIX license generated by the LMF from the `/upgrade` file.

3.10 Cancelling a License

Use the `lmf cancel` command to cancel a license from a specific date. This means that you can stop the use of a product earlier than the day shown by the Key Termination Date field on the PAK. The `lmf cancel` command has the following syntax (see also the `lmf`(8) reference page):

`lmf cancel` *date product* [*producer* [*authorization*]]

The *date* argument can be specified in most common formats, but the order must be: day, month, year. You do not need to use a separator between the day and the month, or the month and the year. For example, 1st July 1990 could be specified as: 1-jul-1990, 1/7/90, 010790, or 1.july.90. To cancel the license for the product ALLSUM on 1st July 1990, you would type:

```
# lmf cancel 1-jul-90 ALLSUM
# lmf load 0 ALLSUM
```

The command does not immediately affect the kernel cache, so you should use the `lmf load` command to update the license for the product in the kernel cache.

You can change the Cancellation Date more than once; you just need to reissue the `lmf cancel` command with a different *date* argument. If you set the Cancellation Date to be after the Key Termination Date shown on the license, the Cancellation Date is ignored.

3.11 Deleting a License

Use the `lmf delete` command to delete a license from the LDB and kernel cache. You can use this command to remove a license from the LDB, when it no longer represents a valid license for the product; for example, if the license has passed its Key Termination Date.

Before you use the `delete` command, you should ensure that you have a copy of the license data in your files. If you delete a license by mistake, you should restore the LDB file (`/usr/var/adm/lmf/ldb`) from a backup, or extract the PAK data from the history file and reregister it.

To delete the license for the product ALLSUM, for example, you should type:

```
# lmf delete ALLSUM
```

Although the command removes the license data from the kernel cache, existing users of the product are allowed to finish using it.

3.12 Updating a License

There are two ways to update a license:

- Modify a license with the `lmf modify` command, if the license has the MOD_UNITS Key Option
- Amend the license with the `lmf amend` command, if you have a Product Authorization Amendment (PAAM) for the product

Note

It is current business policy not to issue Product Authorization Amendments (PAAMs). Do not use the `lmf amend` command unless you have a PAAM.

3.12.1 Modifying a License

Use the `lmf modify` command if you want to change the entry in the Comments field or if you want to change the entry in the Number of Units field on a license with the MOD_UNITS Key Option. For example if the product ALLSUM has the MOD_UNITS Key Option, and you want to increase the number of units on the license from 100 to 200, type:

```
$ lmf modify ALLSUM
```

This displays the current license for ALLSUM with colons (:) before the Comments field and the Number of Units field, for example:

```
                .
                .
        Product Name  ALLSUM
           Producer  DEC

      Number of Units: 100
                .
                .
```

In this example you should use the editor to change the Number of Units from 100 to 200. Changes to lines without colons are ignored.

When you leave the editor, the LMF scans the template to ensure the license has been updated correctly. If it has not, you are given the opportunity to re-enter the editor and correct any mistakes.

When you have successfully modified a license, use the `lmf load` command to copy the modified license into the kernel cache. In this example, type:

```
# lmf load 0 ALLSUM
```

3.12.2 Amending a License

Use the `lmf amend` command when you want to update a license in the LDB after receiving a Product Authorization Amendment (PAAM). A PAAM is used to update an existing license and may only have data in fields that are different to the existing license for the product. A PAAM will always have a different checksum from the existing license. The checksum validates the amended license data; that is, the checksum is generated from the data elements as they appear after the license has been updated with the PAAM data.

Suppose, for example, you already have a license registered for the product ALLSUM and that the license is valid for all versions of the product up to and including Version V2.0. If you wanted to use the license with versions up to and including Version V2.4, you could contact your Digital representative who would arrange for you to be sent a PAAM. The PAAM would contain entries for Version and Checksum (and possibly Product Name to ensure you amend the correct license). To enter the PAAM data into the LDB type:

```
# lmf amend ALLSUM
```

This displays the current license for ALLSUM with colons (:) before the fields that can be changed, for example:

```
                    .
                    .
          Product Name  ALLSUM
              Producer  DEC

       Number of Units: 100

               Version: 2.0
  Product Release Date:
                    .
                    .
```

The current license has the Checksum entry removed, because PAAMs always come supplied with a new checksum. In this example you should use the editor to change the Version from 2.0 to 2.4, and enter the checksum supplied with the PAAM. Changes to lines without colons are ignored.

When you leave the editor, the LMF scans the template to ensure the license has been updated correctly. If it has not, you are given the opportunity to re-enter the editor and correct any mistakes.

When you have successfully amended a license, use the `lmf load` command to copy the amended license into the kernel cache. In this example, type:

```
# lmf load 0 ALLSUM
```

3.13 Monitoring the LDB and Kernel Cache

Use the `lmf list` command to display the details of the registered products on the system. Using `lmf list`, you can:

- Display a summary of all the products registered in the LDB or kernel cache or both
- Display the complete license details for all the products in the LDB or kernel cache or both
- Display the details of specific products only

The `lmf list` command has the following syntax (see also the `lmf`(8) reference page):

`lmf list [ full ] [ source ] [ for product [ producer ] ]`

Use the *full* argument to display the full license details for the product.

Use the *source* argument to choose the source of the license information; there are three choices:

ldb	Displays a summary for each product in the LDB.
cache	Displays a summary for each product in the kernel cache. This shows you the license data that is being checked by the license checking functions.
all	Displays a summary for each product in the LDB, and for each product in the kernel cache.

The following example shows how to display a 1-line summary of all the products registered in LDB:

```
# lmf list

Product                 Status              Users: Total        Active

MESSAGE-ROUTER          active                     unlimited
MR-PROGRAMMER-KIT       disabled
WAN-DEVICE-DRIVERS      enabled
FORTRAN-FOR-ULTRIX      terminated
ALLSUM                  active                     unlimited
ULTRIX                  active                     16           5
```

The Status column indicates the current status of the license. There are six possible license conditions:

- `active`
  The license has been loaded into the kernel cache and can be used to authorize product use.
- `enabled`
  The license has been registered in the LDB but has not been loaded into the kernel cache.
- `disabled`
  The license has been disabled in the LDB.
- `terminated`
  The current date is later than the Key Termination Date specified on the license.

- `cancelled`
  The current date is later than the Cancellation Date specified on the license.
- `multiple`
  This is one of multiple licenses registered for this Product Name and Producer.

The two right-hand columns indicate the amount of product use. For Availability Licensed products, the amount of product use is shown as "unlimited." For Activity Licensed products:

- The Total column shows the maximum number of concurrent users allowed for a product.
- The Active column shows the current number of users of the product.

The following example shows how to display all the license details in the kernel cache for ULTRIX:

```
# lmf list full cache for ultrix

          Product Name: ULTRIX
              Producer: DEC
               Version:
  Product Release Date: 1-JUL-1991
  Key Termination Date:
           Total Units: 160
          Usable Units: 140
       Activity Charge: 10
```

The Total Units field shows the number of license units in the kernel cache. The Usable Units field shows the number of unallocated license units. The Activity Charge field shows the number of license units required for each product user. For Availability Licensed products the Usable Units and Activity Charge fields are zero (0).

3.14 Reviewing Your License Management Activities

Use the `lmf history` to display a list of the `lmf` commands that have been used. The LMF maintains a history file that is a record of the license management operations. The commands recorded in the history file are: `register`, `enable`, `disable`, `issue`, `cancel`, `delete`, `modify`, and `amend`. The creation of a new LDB is also recorded in the history file. For these `lmf` commands, you can:

- Display the history data, which comprises product name, date and time of the command, and the fields that were changed on the license
- Display the history data and the license (as it appeared before the command was issued)
- Display a 1-line summary of the history data for each command issued
- Display the history data for commands issued after a certain date
- Display the history data for specific products

The `lmf history` command has the following syntax (see also the `lmf`(8) reference page):

`lmf history [` *length* `] [` from *date* `] [` for *product* `[` *producer* `] ]`

Use the *length* argument to choose the length of the history data for each command. There are two choices:

short Displays a 1-line summary of the history data for each command issued.

full Displays the history data for each command issued, and the license as it appeared before the command was issued.

Use the `from` *date* argument to display the data for each command issued after the *date* specified. The *date* argument can be specified in most common formats, but the order must be: day, month, year. You do not need to use a separator between the day and the month, or the month and the year. For example, 1st March 1989 could be specified as: 1-mar-1989, 1/3/89, 010389, or 1.march.89.

The following example shows how to display the history data for one product (ALLSUM):

```
# lmf history for ALLSUM

Product Name: ALLSUM
Producer    : DEC
Command     : ENABLE
Date        : 26-FEB-1989
Time        : 12:02:32

Product Name: ALLSUM
Producer    : DEC
Command     : DISABLE
Date        : 15-JAN-1989
Time        : 11:57:26

Product Name: ALLSUM
Producer    : DEC
Command     : REGISTER
Date        : 4-NOV-1988
Time        : 11:54:15
```

The next example shows the 1-line summary of the history data for the same set of `lmf` commands:

```
# lmf history short for ALLSUM

Product Name      Producer          Command    Date        Time

ALLSUM            DEC               ENABLE     26-FEB-1989 12:02:32
ALLSUM            DEC               DISABLE    15-JAN-1989 11:57:26
ALLSUM            DEC               REGISTER   4-NOV-1989  11:54:15
```

3.15 Changing the Number of Active CPUs

When a system is rebooted, the LMF checks the maximum possible number of active CPUs on the system, and uses this value to determine the System Marketing Model. The SMM is used by some products to define the number of license units needed in the kernel cache before access to the product is granted.

If you change the number of active CPUs the SMM may change, and so may the number of license units needed in the kernel cache to access a product. Use the `lmf reset cpus` command to determine a new SMM.

This section describes the license management actions you should take if you change the number of active CPUs:

- For system maintenance purposes
- To reduce the license unit requirement of the system

3.15.1 System Maintenance

If you reduce the number of active CPUs for system maintenance purposes, you do not need to take any special license management actions. The LMF continues to use the current SMM, even though the number of active CPUs has changed.

When you return to the original number of active CPUs, the LMF continues to use the current SMM, which now accurately reflects the number of active CPUs. Again, you do not need to take any license management actions.

3.15.2 Reducing the License Unit Requirement

You can reduce the license unit requirement of your system by reducing the number of active CPUs on the system. For example, assume you have reduced the number of active CPUs from two to one. To determine the new SMM for the system, you would type:

```
# lmf reset cpus
```

Before you return to the original number of active CPUs, you must determine the new SMM. For example, assume you are ready to increase the number of active CPUs from one to two. To determine the new SMM for the system, you would type:

```
# lmf reset cpus 2
```

If you do not determine the new SMM before returning to the original number of active CPUs, the LMF will prevent any further access to the licensed products, although existing users will be able to finish using them.

3.16 Managing Availability Licensed Products

Before you order a PAK, you should define your software and hardware requirements to your Digital representative so that you get a license of the correct size. For availability licensed products, the license you register in the License Database should provide enough license units to allow full access to the licensed product. For example, the PAK for a software product to be used on a processor which requires 400 license units should specify 400 license units.

Sometimes, users with multiple stand-alone systems cannot match their licenses to meet every circumstance. For example, you may manage two stand-alone processors; VAXBIG, which requires a 700-unit license, and VAXMID, which requires a 400-unit license. If you get a 700-unit license for VAXBIG, you can move that license (with an `lmf issue` command) to VAXMID when you shut down VAXBIG for a memory upgrade. You may not, however, be able to move and register a license intended for VAXMID to VAXBIG.

3.17 Providing More Availability License Units

Sometimes you may need to provide more license units than are currently registered in the LDB for the product. In the previous example, the 400-unit license did not provide enough product availability for VAXBIG, which required 700 units. If the license has the MOD_UNITS Key Option, and you need to move the license to VAXBIG, increase the number of units on the license to 700 using the `lmf modify` command. If the license does not have the MOD_UNITS Key Option, and you need to move the license to VAXBIG, contact a Digital representative, who will probably recommend one of the following:

- A new license that provides at least 700 license units.
- Another license for the same product that provides at least an additional 300 license units. If the terms of your license contract allow it, you can register the two licenses, allowing the LMF to combine the license units and producing the equivalent of a 700-unit license. This can authorize the product on VAXBIG. For a complete explanation of license combination, see Section 3.20.
- A Product Authorization Amendment (PAAM) that increases your current license to at least 700 units.

3.18 Managing Activity Licensed Products

As with availability licensed products, you should define your software and hardware requirements to your Digital representative so that you get a license of the correct size. The Activity License you register in the LDB should provide enough license units to allow some predetermined number of processes access to the product.

For example, if a software product requires 25 license units per activity on your processor, and PAKs come in 4-activity increments, your license may provide units in a multiple of 100: 100, 200, 400, and so forth. Note that a 120-unit license would provide no more use than a 100-unit license on such a processor.

Because processors can have different license unit requirements per activity, the number of users authorized by a license can vary according to the processor used. For example, you may manage two stand-alone processors; VAXBIG which requires 25 license units per activity to authorize a product, and VAXMID which requires only 20 license units per activity to authorize a product.

If you obtain a 125-unit Activity License for VAXBIG, you can temporarily move that license (using the `lmf issue` command) to VAXMID when you shut down VAXBIG for maintenance. The 125-unit license, which provided five users product access on VAXBIG, provides six users product access on VAXMID. This provides the backup you need. Also, unlike the situation with Availability Licenses, you can move a 4-user 80-unit license originally for VAXMID to VAXBIG. On VAXBIG the license provides access to only three users, however.

As with Availability Licenses, you can successfully register a license in the License Database that a user cannot successfully activate. If you register a 40-unit license that provides product access to two users on a MicroVAX II, the same license may not allow access to one user on a VAX 8800 that might require 50 units per access.

3.19 Providing More Activity License Units

If the license has the MOD_UNITS Key Option, you can increase the number of units on the license using the `lmf modify` command. If the license does not have the MOD_UNITS Key Option, contact a Digital representative, who will probably recommend one of the following:

- A new license with more units.
- Another license for the same product that provides additional license units. If the terms of your license allow it, you can register two or more licenses for the same product and combine them to form one larger license. For a complete explanation of license combination, see Section 3.20.
- A different kind of license. Because some products offer both Activity and Availability Licenses, a change to an Availability License may be recommended.
- A Product Authorization Amendment (PAAM) that increases the size of your current license.

3.20 Combining Licenses

Combining licenses means registering two or more licenses for the same product in the License Database, and loading them into the kernel cache to form a single license. Licenses which have the NO_SHARE Key Option cannot be combined.

The following fields must be the same on the original licenses:

- Issuer
- Product Name
- Producer
- Hardware-Id
- Product Token

Note

The Authorization Number must be different on the licenses; the LMF does not allow the same license to be registered more than once in the same LDB.

Register the licenses in the LDB in the usual way; that is, using the `lmf register` command (see Section 3.2). The licenses registered appear as separate entries in the LDB. To combine licenses to form a single license in the kernel cache, use the `lmf load` command (see Section 3.5.1). For example, if you have registered two licenses for the product ALLSUM, you can form a single license in the kernel cache by typing:

```
# lmf load 0 ALLSUM
```

The combined license appears as one entry in the kernel cache.

The Number of Units for the license is the total number supplied by the original licenses. For example, if the Number of Units entries on the original licenses were 300 and 400, the combined license would be a 700-unit license.

The Product Release Date and the Key Termination Date for the combined license are the earlier of those supplied by the original licenses. For example, if the Product Release Dates for the licenses were 1-JAN-1990 and 1-AUG-1990, the combined license would have a Product Release Date of 1-JAN-1990.

The Version for the combined license is the lower of those supplied by the original licenses. For example, if the Version for the original licenses were V1.2 and V1.4, the combined license would have a Version of V1.2.

To remove the combined license from the kernel cache, use the `lmf unload` command (see Section 3.6).

Error Messages A

This Appendix lists and explains the error messages you may encounter. The error messages are listed in alphabetical order, and where appropriate a course of action is recommended to correct the error.

A.1 Accessing Licensed Software

You may encounter the following messages when attempting to access software that provides full support for the LMF.

Attempted usage exceeds active license units

You have tried to access an activity licensed product, but the number of units available is less than the license unit requirement for the current processor. This means that the maximum number of simultaneous users of the product has been reached.

Contact your system manager to find out if the maximum number of simultaneous users can be increased. Alternatively, wait until the number of users of the product falls below the maximum and try to access the product again.

License is invalid for this version of the product

You have tried to access a product but the version number on the license is lower than the product version.

Contact your system manager. Your system manager may need to install an earlier version of the product or contact Digital for a new PAK.

No license found for this product

You have tried to access a product that does not have a valid license in the kernel cache.

Contact your system manager.

A.2 Using the lmf(8) Utility

You may encounter the following messages when using the `lmf` utility.

A license for *product* cannot be disabled

You have tried to disable a license for the *product* specified. Certain licenses cannot be disabled, for example, the license created by the LMF from information in the `/upgrade` file, see Section 3.3.

A license for *product* cannot be issued

You have tried to issue a license for the *product* specified. Certain licenses cannot be issued, for example, the license created by the LMF from information in the `/upgrade` file, see Section 3.3.

A license that has been cancelled cannot be enabled

You have tried to enable a license that has passed its cancellation date.

If you want to enable the license, you should change the cancellation date, see Section 3.10.

A license that has terminated cannot be enabled

You have tried to enable a license that has passed its termination date. A license that has passed its termination date should be deleted from the LDB, see Section 3.11.

Activity charge has changed - reboot to load new license for *product producer*

The activity charge has changed for the *product* specified. This may have happened if the license type has changed, for example, from an activity to availability license, or if the SMM has changed.

Reboot your system to fully reset the kernel cache.

"Activity Table Code" amended - protected field

You have changed the Activity Table Code. The Activity Table Code can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Activity Table Code" - invalid format

The Activity Table Code you have entered for the license either does not match a valid License Unit Requirement Table code or is not of the form "CONSTANT=*integer*."

When you register the license, you must enter the Activity Table Code exactly as it appears on the PAK.

"Activity Table Code" missing from PAK entry

You have not entered an Activity Table Code for the license.

When you register the license, you must enter all the data from the PAK.

Ambiguous command *string*

The *string* used as an abbreviation for a command was ambiguous.

When you type in a command, you must use enough letters to distinguish it from other commands.

"Authorization Number" amended - protected field

You have changed the Authorization Number. The Authorization Number should never be changed, as it helps to uniquely identify each license.

"Authorization Number" missing from PAK entry

You have not entered an Authorization Number for the license.

When you register the license, you must enter all the data from the PAK.

"Availability Table Code" amended - protected field

You have changed the Availability Table Code. The Availability Table Code can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Availability Table Code" - invalid format

The Availability Table Code you have entered for the license either does not match a valid License Unit Requirement Table code or is not of the form "CONSTANT=*integer*."

When you register the license, you must enter the Availability Table Code exactly as it appears on the PAK.

"Availability Table Code" missing from PAK entry

You have not entered an Availability Table Code for the license.

When you register the license, you must enter all the data from the PAK.

Cannot unload this many users

You have specified too many users with the `lmf unload` command.

Reissue the `lmf unload` command with fewer users. To remove all the license units for the product from the kernel cache, you should specify zero (0) as the number of users.

"Checksum" amended - protected field

You have changed the Checksum. The Checksum can be changed only when using `lmf amend` with a suitable Product Authorization Amendment.

Checksum does not validate

When you attempted to register a license, the checksum did not validate the license information you entered. The checksum contains, in encrypted form, all the license information from the PAK. If you enter inaccurate license information, you receive this message.

Carefully review all licensing information on the PAK. When you register the license, you must enter all the information exactly as it appears on the PAK.

"Checksum" missing from PAK entry

You have not entered a Checksum for the license.

When you register the license, you must enter all the data from the PAK.

Combine *product authorization_number* **with** *product authorization_number*

The two licenses shown have been combined to form a single license in the kernel cache.

Error adding to kernel cache

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing LURT file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing temporary file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error closing the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error creating license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error creating the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error determining SMM

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error locking license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error locking the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error no entry in LURT for this SMM

The LURT version and the SMM version are inconsistent.

Contact your Digital representative.

Error opening file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error opening license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error opening LURT file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error opening temporary file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error opening the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error reading kernel cache

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error reading license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error reading LURT file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error reading the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error renaming temporary file *filename* **to license database** *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error setting the number of cpus

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error unlocking license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error unlocking the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error updating kernel cache

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error writing to license database *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error writing to the history file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Error writing to temporary file *filename*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Failed to create process for editor

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

"Hardware-Id" amended - protected field

You have changed the Hardware-Id. The Hardware-Id should only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

History file locked - retrying ...

You have tried to access the history file at the same time as another user. The `lmf` utility automatically grants you access to the history file as soon as the other user has finished with it.

Information provided was ambiguous; multiple licenses were found

You did not provide enough information for the command to identify one license for the product.

If you have more than one license with the same Product Name, you can distinguish them by specifying the Producer and the Authorization Number.

Internal LMF error was encountered

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Invalid argument *string*

The *string* specified was not recognized as a valid argument for the command.

For a complete description of the syntax of the `lmf` commands, see the `lmf`(8) reference page.

Invalid entry for availability/activity table code for *product producer*

The LURT version and the LMF utility are inconsistent.

Contact your Digital representative.

Invalid LURT entry for *product producer*

The LURT version and the PAK information are inconsistent.

Contact your Digital representative.

"Issuer" amended - protected field

You have changed the Issuer. The Issuer can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Issuer" missing from PAK entry

You have not entered an Issuer for the license.

When you register the license, you must enter all the data from the PAK.

"Key Options" amended - protected field

You have changed the Key Options. The Key Options can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Key Termination Date" amended - protected field

You have changed the Key Termination Date. The Key Termination Date can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

License already registered

You have tried to register a license that is already in the LDB.

License database locked - retrying ...

You have tried to access the LDB at the same time as another user. The `lmf` utility automatically grants you access to the LDB as soon as the other user has finished with it.

License too small to load this many users

You have specified too many users with the `lmf load` command.

Reissue the `lmf load` command with a lower number of users. To load all the license units for the product into the kernel cache, you should specify zero (0) as the number of users.

License unchanged

You have left the editor after a `lmf amend` or `lmf modify` command, without making any changes to the existing license.

Missing arguments

You have not specified enough arguments with the command.

For a complete description of the syntax of the `lmf` commands, see the `lmf`(8) reference page.

Multiple licenses could not be combined for *product producer*

You have tried to combine licenses, at least one of which has the NO_SHARE Key Option. The LMF will not combine licenses if any of the following fields are different: Issuer, Product Name, Producer, Product Token, Hardware-Id. Only licenses without the NO_SHARE Key Option can be combined.

No entries in license database

You have tried to list the contents of the LDB, but the LDB is empty.

No entry in the history file for this product

You have specified a product name with the `lmf history` command, but no `lmf` commands have been recorded for this product. The history file records only the following `lmf` commands: `register`, `enable`, `issue`, `cancel`, `delete`, `modify`, and `amend`.

When you use the *product* argument, it should be specified exactly as it appears on the PAK. If you use the `from` *date* argument, ensure that the date you specify is not later than the date of the last command recorded for the product.

No entry in the kernel cache for this product

The product name you have specified in the command does not have an entry in the kernel cache.

When you use the *product* argument, it should be specified exactly as it appears on the PAK.

No entry in the license database for this product

The product name you have specified in the command does not have an entry in the LDB.

When you use the *product* argument, it should be specified exactly as it appears on the PAK.

No valid license was found for this product

You have tried to load a nonvalid license for a product.

Ensure that you entered the correct product name, and that the license is not terminated, disabled, or cancelled.

Not enough memory

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

Not enough units to load *product producer*

You have tried to load a license for the product specified, but the license does not have enough license units for the current SMM.

For a full description of how to provide more license units, see Section 3.17 and Section 3.19.

"Number of Units" amended - protected field

You have changed the Number of Units. The Number of Units can be changed either when using the `lmf modify` command on a license which has the MOD_UNITS Key Option, or when using `lmf amend` with a suitable Product Authorization Amendment.

PAK not registered

You have left the editor after a `lmf register` command without saving the file.

Permission denied

You have tried to execute a `lmf` command but are not logged into the system as the superuser (root login).

Log in to the system as superuser and reissue the command.

"Producer" amended - protected field

You have changed the Producer. The Producer should never be changed, as it helps to uniquely identify each license.

"Product Name" amended - protected field

You have changed the Product Name. The Product Name should never be changed, as it helps to uniquely identify each license.

"Product Name" missing from pak entry

You have not entered a Product Name for the license.

When you register the license, you must enter all the data from the PAK.

"Product Release Date" amended - protected field

You have changed the Product Release Date. The Product Release Date can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

"Product Token" amended - protected field

You have changed the Product Token. The Product Token can only be changed when using `lmf amend` with a suitable Product Authorization Amendment.

The kernel cache is empty

You have tried to list the contents of the kernel cache, but it is empty.

The license database file *filename* is corrupt - restore most recent backup

The LDB file has been corrupted by some means and cannot be read by the `lmf` utility.

The license database is incompatible with this version of lmf

The LDB version and the `lmf` utility are inconsistent.

Contact your Digital representative.

The LURT file *filename* is corrupt - restore most recent backup

The LURT file has been corrupted by some means and cannot be read by the `lmf` utility.

Unrecognized cpu for *product producer*

A system error has occurred, causing the `lmf` utility to exit, with a nonzero error status.

"Version" amended - protected field

You have changed the Version. The Version can be changed only when using `lmf amend` with a suitable Product Authorization Amendment.

Warning creating new history file

The `lmf` utility has not found an existing history file and is creating a new one.

Warning creating new license database

The `lmf` utility has not found an existing LDB and is creating a new one.

Glossary

This glossary defines a number of terms and acronyms that may be encountered.

activity license

A license which defines the number of concurrent users allowed access to a product. For example, a 4-activity license can have enough license units to allow four users to access the product simultaneously.

authorization number

The unique number assigned by the PAK issuer to a specific PAK. The PAK issuer name and authorization number identify a license.

availability license

A license that makes a product available to all the users of a system. The LMF makes a product accessible when the number of license units on a license matches or exceeds the license unit rating of the current processor.

CDROM

See the entry for compact-disc read-only memory (CDROM).

checksum

An encoded number calculated from the other information supplied with a PAK. The checksum is used by the LMF to validate the rest of the PAK data. The checksum string always begins with a number, which is the only number in the string. The other sixteen positions are always alphabetic characters from A to P.

compact-disc read-only memory (CDROM)

A media for the consolidated distribution of software. ULTRIX operating systems and layered products can be distributed on single CDROMs, with software access authorized by PAKs and the LMF.

DDSLA

See the entry for Digital Distributed Software Licensing Architecture (DDSLA).

Digital Distributed Software Licensing Architecture (DDSLA).

This engineering architecture is based on the concept of license units as an abstract mechanism for counting and sizing customer computing environments.

key termination date

A field on a PAK that defines when a license contract is no longer valid, that is, when the LMF no longer authorizes product use.

LDB

See the entry for License Database (LDB).

license amendment

Updating an existing license by entering data from a Product Authorization Amendment (PAAM) in the License Database.

license combination

Using the license units from two or more licenses for the same product to provide more product access. Two licenses each with 100 units combine to equal a 200-unit license. Licenses that specify the NO_SHARE option cannot be combined.

License Database (LDB)

A system file which contains the licenses registered on the system. The LMF also maintains a kernel cache with the license information in it, and this is used by the LMF to prevent unlicensed product use.

License Management Facility (LMF)

Part of ULTRIX operating systems that enables the on-line management of software license data, and also helps prevent accidental unlicensed use of software.

license registration

The task you perform when you enter license data from a Product Authorization Key into the License Database. To register a license, use the `lmf register` command.

license unit

The basic unit of measurement that Digital uses to specify how much product use a license provides. Digital gives each license intended to be used with LMF a size, specified in license units. For example, a license can be a 50-unit license, a 20-unit license, or a 700-unit license.

License Unit Requirement Tables (LURTs)

A table provided by Digital as part of ULTRIX operating systems that specifies a series of license unit requirements, essentially performance ratings, for each System Marketing Model. Processors that provide more performance (other ratings may be unrelated to performance) have greater license units requirements.

LMF

See the entry for License Management Facility (LMF).

LURT

See the entry for License Unit Requirement Tables (LURTs).

PAAM

See the entry for Product Authorization Amendment (PAAM)

PAK

See the entry for Product Authorization Key (PAK).

PAK identification

The Product Authorization Key issuer name and the authorization number. Together, they uniquely identify a license.

PAK issuer

The company that creates the license contract for the software. The PAK issuer name and license authorization number uniquely identify a license. PAK issuers are usually the same as software producers but can operate under agreement with the producer.

Product Authorization Amendment (PAAM)

Provides information to amend the license for an existing licensed software product. Without a current PAK or the appropriate PAAM, you may not be able to use an installed software product. A PAAM contains a unique authorization checksum and the information needed to amend current license information.

Product Authorization Key (PAK)

A list of essential information about a software license that must be registered in the License Database in order to use a product. It is produced by a PAK issuer and delivered to you by mail, electronic transfer, or by telephone.

Product Identification

The software producer name and product name. Together they uniquely identify a software product for licensing.

SMM

See the entry for System Marketing Model (SMM).

software license

A contract between a license producer (Digital) and a license receiver (customer) that grants permission to use a specific software product as described

by the applicable Software Product Description (SPD), and the terms and conditions of the license contract. A PAK supplies the information that results from a software license contract.

SPD

See the entry for Software Product Description (SPD).

Software Product Description (SPD)

The legal document that describes the software product. This document contains the precise product release level that comprises product version and official product release date.

System Marketing Model (SMM)

The model name of a computer system, as used in marketing and pricing. The SMM generally corresponds to the name on the front panel of the processor cabinet. The LMF uses this value rather than hardware CPU-type because different marketing models may use the same CPU with different pricing and licensing rules.

Index

M

N

O

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to
Software Licensing
AA-PBKRA-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to
Software Licensing
AA-PBKRA-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

digital

ULTRIX

Guide to System Environment Setup

Order Number: AA-ME89B-TE
June 1990

Product Version: ULTRIX Version 4.0 or higher

digital equipment corporation
maynard, massachusetts

Contents

About This Manual

1 Modifying System Files

Tables

About This Manual

The objective of this guide is to provide you with information needed to set up and maintain a working environment for users at your site. The guide assists you as you set up your particular environment and presents guidelines from which you can develop specific procedures for your site.

Audience

The *ULTRIX Guide to System Environment Setup* is written for the general user or person responsible for managing and maintaining an ULTRIX system. It assumes that this individual is familiar with ULTRIX commands, the system configuration, the system's controller/drive unit number assignments and naming conventions, and an editor such as `vi`(1) or `ed`(1). You do not need to be a programmer to use this guide.

Organization

This manual consists of 6 chapters, one appendix, and an index. The chapters and the appendix are:

Chapter 1	Modifying System Files Identifies and describes the most commonly accessed ULTRIX system files and defines the file formats. The chapter also identifies system utilities that you can use to modify the files.
Chapter 2	Adding Devices Describes how to add devices using the `MAKEDEV` script.
Chapter 3	Managing the Print System Explains how to set up the print system manually.
Chapter 5	Adding and Deleting Software Subsets Explains how to use the `setld` utility to add, delete, load, and verify software subsets.
Chapter 6	Monitoring and Managing System Performance Offers guidelines for monitoring system performance.
Appendix A	Device Mnemonics Lists the supported device mnemonics and explains how to obtain detailed reference page information on devices.

Appendix B Support of the HSC/CI Hardware
Describes the relationship between the CI (dual path bus) and the I/O subsystems (HSCs).

Related Documents

The following manuals provide additional informational on the topics in this manual:

- *ULTRIX Guide to System and Network Setup*
- *ULTRIX Security Guide for Administrators*
- *ULTRIX Release Notes*

You should also have the hardware documentation for your system and peripherals.

Conventions

The following conventions are used in this manual:

#	A number sign is the default superuser prompt.
`user input`	This bold typeface is used in interactive examples to indicate typed user input.
`system output`	This typeface is used in interactive examples to indicate system output and also in code examples and other screen displays. In text, this typeface is used to indicate the exact name of a command, option, partition, pathname, directory, or file.
UPPERCASE lowercase	The ULTRIX system differentiates between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function definitions must be typed exactly as shown.
rlogin	In syntax descriptions and function definitions, this typeface is used to indicate terms that you must type exactly as shown.
filename	In examples, syntax descriptions, and function definitions, italics are used to indicate variable values; and in text, to give references to other documents.
[ ]	In syntax descriptions and function definitions, brackets indicate items that are optional.
. . .	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
`cat`(1)	Cross-references to the *ULTRIX Reference Pages* include the appropriate section number in parentheses. For example, a reference to `cat`(1) indicates that you can find the material on the `cat` command in Section 1 of the reference pages.

Modifying System Files 1

The ULTRIX system files perform a number of functions. For example, they enable log ins, the setup of mail aliases, and the display of a login message. Most system files are created during the system installation; however, you can create or modify files after the system has been installed.

This chapter contains descriptions, sample entries, and instructions for modifying the following system files:

- Password File (`/etc/passwd`)
- Group File (`/etc/group`)
- Terminal Initialization File (`/etc/ttys`)
- File System Table (`/etc/fstab`)
- Mail Aliases File (`/usr/lib/aliases`)
- Clock Table Daemon (`/usr/lib/crontab`)
- Message-of-the-Day File (`/etc/motd`)

1.1 The Password File

The password file, `/etc/passwd`, is a data file that contains an entry for every user who has login privileges on your system. Each entry in the `/etc/passwd` file has fields that specify the following information:

- User login name
- User password
- User identification number
- Group identification number
- A description of the user
- The pathname of the user home directory
- The pathname of the default shell or command to be executed immediately following login

Each entry in the `/etc/passwd` file must contain at least the following fields: name, encrypted password, user identification number, and group identification number. Entries that do not include the preceding fields are ignored, or may introduce a security problem on some systems. See the *ULTRIX Security Guide for Administrators* for more information.

Each field in the password file is separated by colons. The format for each entry is as follows:

name:[password]:user-id:group-id:[description]:home-directory:[shell]

name — The first field contains the user's login name (1 to 8 characters). The system uses this name to establish login permission.

password — The second field contains the user's encrypted password. An entry in this field is optional. The system uses this password to verify login permission.

user-id — The third field contains the user's identification number (user ID). The system uses this number to determine a user's identity. Once login permission is granted, the system internally translates the login name to this user ID number and uses it to identify the user's processes and to determine owner access permission to files. This number must be unique for each user and should be less than 32000.

group-id — The fourth field contains the primary group identification number (group ID). The system uses this number to determine a user's default group classification. You can set up the permissions for any file so that users with the same *group-id* numbers can access the file, but those users with different *group-id* numbers cannot. Once login permission is granted, the system internally establishes the user's group ID number and uses it in determining group access permission to files. This number must be unique for each group and should be less than 32000.

A user may belong to a maximum of 32 groups. The `/etc/passwd` file lists only the user's primary group. You can see the user's secondary groups by displaying the `/etc/group` file.

description — The fifth field contains additional user information. For example: user name, office location, office phone number, and home phone. The user name can be an ampersand (&), which means that the login name and the user name are the same. User's can maintain the *description* field with the `chfn` command. See `chfn`(1) for more information.

home-directory — The sixth field contains the absolute pathname to the user's home (initial working) directory. After establishing the appropriate user and group identification, the system uses this pathname to place the user in the named directory.

shell — The seventh field contains the absolute pathname to the command that is to be executed immediately upon conclusion of the login process. This is normally a version of the shell (command interpreter) such as `/bin/csh` or `/bin/sh`. It can also be used to allow the user limited access to the system. For example, by replacing a shell version with the full pathname of a particular command, you can log in using the name of the command. The shell will then run the command and log the user out once the command has been executed.

An entry in this field is optional. If nothing is specified, the system automatically invokes `/bin/sh`.

Following is a sample password file:

```
root:R,r97fsje2oss:0:1:System PRIVILEGED Account:/:/bin/csh
field:Pa9rek3.1l5e:0:1:F S PRIVILEGED Account:/usr/field:/bin/csh
operator:sruF3.9ir,ePw:0:28:Operator PRIVILEGED Account:/opr:/opr/ops
guest:n3Rel9s22:10:33:Guest account:/tmpguest:/bin/date
jjd::34:10:John Joseph Doe:/usr/staff/jjd:/bin/csh
jws::24:10:John Walter Smith:/usr/staff/jws:/bin/csh
```

If you are using the Yellow Pages service on your system, see the *ULTRIX Guide to Yellow Pages Services* for more information on the password file. While the format of the file is similar, differences exist. If you are using the BIND service, see the *ULTRIX Guide to the BIND Service*.

1.1.1 Modifying the Password File

The system automatically creates a generic `/etc/passwd` file during the installation, but you can modify this file to include site and user specific information. By default, the `/etc/passwd` file is read only. Only the superuser can modify any field in the password file using system commands. Once logins are enabled, registered users can modify only their password, description, and shell fields using system commands.

Note

Avoid using a text editor to edit the password file. A text editor does not perform the necessary processing and locking to keep your password file secure. Use the `vipw` command to edit the password file.

The following commands enable you to modify the password file while performing the necessary protection and locking of the file:

`vipw`	Enables the superuser or root to edit any field in the password file.
`adduser`	Adds new accounts to the password file.
`removeuser`	Removes accounts from the password file.
`passwd`	Enables users to change the password field.
`chsh`	Enables users to changes the login shell field.
`chfn`	Allows users to change the description field.

The following sections discuss these commands in detail. For additional information, see the *ULTRIX Reference Pages*.

1.1.1.1 Editing the Password File – To ensure proper locking and processing of the password file, use the `vipw` command to edit the password file. The `vipw` command enables you to edit any field in the password file. You must have root privileges to use the `vipw` command.

To use the `vipw` command, type the following:

```
% vipw passwd
```

The vipw command invokes the vi editor unless the environment variable EDITOR indicates an alternate editor. If the password file is being modified by another user, the following message is displayed:

```
vipw: password file busy
```

When you exit the editing session, the vipw command performs a number of consistency checks on the password entry for root, and does not enable a password file with a corrupted root entry to be installed.

1.1.1.2 Adding User Accounts to the Password File – The adduser command is an interactive facility that enables you to create accounts for new users on the local system. In addition to adding new users to the password and group files, the adduser command sets up a home directory with the generic startup files and creates a bin subdirectory.

To invoke the adduser command, type the following at the system prompt:

```
% /etc/adduser
```

See the *ULTRIX Guide to System and Network Management* for step-by-step instructions on using the adduser command. If you are using the Yellow Pages service, see the *ULTRIX Guide to Yellow Pages Services* for information on adding users. If you are using the BIND services, see the *ULTRIX Guide to the BIND Service*. Additionally, the *ULTRIX Security Guide for Administrators* contains detailed information on the passwd file and the /etc/auth database.

1.1.1.3 Removing User Accounts from the Password File – The removeuser command is an interactive facility that removes user accounts from the password file, and optionally deletes the user's home directory and files. This command does not alter the group file; hence, you must edit the /etc/group file to remove a user from a group. For more information on the group file, see Section 1.2.

Use the following steps to remove a user's account:

1. Invoke the removeuser command:

   ```
   # /etc/removeuser
   ```

2. Type the user's login name:

   ```
   Enter login name for user to be removed: tippet
   ```

   In this example, the login name tippet is entered. The removeuser command searches the password file for the user name.

   If the user name exists, the password entry for that user name is displayed. For example:

   ```
   This is what the entry in /etc/passwd looks like:

        tippet::543:15:Carl A. Tippet:/usr/user1/tippet:/bin/csh
   ```

   If the user name is incorrect, the command facility displays a message stating that the user does not exist in the /etc/passwd file, then exits.

3. Indicate whether or not you want to delete the displayed entry:

```
Is this the entry you wish to delete?  y
Working ...

User tippet removed.
```

If you type **y** for yes, as in this example, the user account is removed. If you type an **n** for no, the command facility returns you to the system prompt and the password entry remains unchanged.

4. Determine whether you want to remove the user's home directory, subdirectory, and files:

```
Do you want to remove tippet's home directory,
all subdirectories and files (y/n)?  y

You should have backed up tippet's files if you do not wish to lose them.

Are you sure that you want to remove tippet's files (y/n)? y

Deleting /usr/user1/tippet

#
```

If you type **y** for yes, the command facility verifies that you want to remove the files, as in the previous example, then returns you to the system prompt. If you type **n** for no, the command facility saves the files, and returns you to the system prompt. The `removeuser` session is complete at this time.

1.1.1.4 Changing the User Password – If you are logged in as root or the superuser, you can change any users' password field in the password file using either the `vipw` command or the `passwd` command. After logins are enabled, registered can change their own password fields using the `passwd` command only.

The `passwd` command changes or adds a password field. A user password must contain between 6 and 16 characters comprised of at least 3 different characters. For example, the password `ababab` is not acceptable. To use this command facility, invoke the password facility, type the old password and the new password, then retype the new password verify its accuracy as follows:

```
% passwd
Old password:
Enter new password:
Verify:
```

The password is not echoed to the terminal screen for security reasons. If you use the `passwd` command on a hardcopy terminal, dispose of the printout when you have completed your session. If you are using the Yellow Pages Service, see the *ULTRIX Guide to Yellow Pages Services* for information on changing the user password.

1.1.1.5 Changing the Login Shell – To change the login shell field listed in the `/etc/passwd` file, use the `chsh` command. This command checks the password file for your current login shell, displays the current shell, and allows you to type a different shell. For example:

```
% chsh
Changing login shell for tippet
Shell [/bin/csh]: sh
```

If your current shell or the new shell is not listed in the `/etc/shells` file, the entry remains unchanged.

1.1.1.6 Changing the Description Field – To change the description field in the password file, you can use the `chfn` command. The description field in the password file is used by the `finger` command, and other programs. This command facility prompts you for the following information:

- User name
- Office location
- Office number
- Home number

The command facility displays defaults for each entry in brackets. To accept the default, press the RETURN key. To leave an entry blank, type **none** at the prompt. Entries in the description field cannot contain colons, commas, or control characters; however, phone numbers may be entered with or without hyphens. To invoke this command facility, type the following:

```
% chfn
Changing finger information for tippet
Name [Carl A. Tippet]: <RET>
Office number [GRR-13/DH]:
MMM-11/DH
Office phone []:
666-8888
Home phone [555-5555]:
none
```

If you are logged in as the superuser or root, you can change another users description by specifying the command and user's login name on the same command line. For example:

```
% chfn loginname
```

1.2 The Group File

The system group file, `/etc/group`, contains data for groups and group members. This data file allows users with different group identification numbers (group IDs) to access common files. The system uses the `/etc/group` file to establish access permissions to files created specifically by group members. A user can be assigned to a maximum of 32 groups. Each entry in the `/etc/group` file contains four fields. Each field is delimited by a colon, and the items in the fourth field are further delimited from each other by commas. A group file entry cannot exceed 1024 characters in length. The format of the `/etc/group` file is as follows:

group:password:group-id:name,name...

group The name of the group.

password The encrypted password. While this field is not used, place an asterisk (*) in this field to eliminate group password matching. When creating a new entry, put an asterisk (*) in this field. The asterisk eliminates group password matching.

group-id The group identification number. The system uses this number to determine group access permissions to files. The users' primary *group-id* is listed in the `/etc/passwd` file. The users' secondary groups is listed in the `/etc/group` file. The *group-id* must be unique and should be less than 32000.

name The login names of the current group members. Each name is delimited from the other by a comma. A group file entry can have as many as 200 members, provided that the entire entry does not exceed 1024. Member names can be continued on the next line.

The following example shows the contents of a group file:

```
clowns:*:25:brown,green,white
tigers:*:53:austin,wake,martinez
angels:*:47:howell,baskerville,tillman,wake
```

For more information on the group file, see the `group`(5) reference page in the *ULTRIX Reference Manual.*

1.2.1 Modifying the Group File

To modify the group file, `/etc/group`, you can use a standard text editor or, if you are adding only a new group, you can use the `addgroup` command. The next two sections discuss how to edit the group file using a standard text editor and how to use the `addgroup` command.

Note that the `adduser` command also allows you to add new groups to the group file when you are creating new user accounts in the password file. For more information on the `adduser` command, see Section 1.1.1.2.

1.2.1.1 Editing the Group File – By editing the group file using a standard text editor, you can do the following:

- Add new users.
- Create a new group entry.
- Remove a member from a group.
- Remove a group entry.

To add a new entry to the group file, open the file with an editor, and add the information that defines the new group. For example, if you wanted to add a group called diners, the entry may be created as follows:

```
diners:*:79:lapin,johnson,howell
```

The group name is diners, the password field is marked with an asterisk (*) to eliminate password matching, the group id is 79, and the members of the group are lapin, johnson, and howell.

To add a new user to the group file, edit the `/etc/group` file and add the new name to the fourth field of the group entry. A comma must separate each entry in the fourth field. For example, if you added **ellis** to the diners group, the entry would appear as follows:

```
diners:*:79:lapin,johnson,howell,ellis
```

To delete an existing group from the /etc/group file, open the file with an editor and remove the entry that defines the group you want removed. To remove a current member from a particular group, open the file with an editor, and delete the member's login name and delimiting comma from the group entry.

1.2.1.2 Adding a Group to the Group File – To add groups to the group file, you can either use a text editor as described in Section 1.2.1.1 or you can use the addgroup command. The addgroup command is an interactive facility that lets you add a new group and group id to the password file. The new group name must be unique to the group file. The addgroup command automatically checks the group ids and displays the next available group id in brackets. To accept the default group id, press the RETURN key. For example:

```
Enter group name for new group: moo

Enter group id for the new group [79]: <RET>
```

Using the information from the previous example, a new entry would be created as follows:

```
moo:*:79
```

To add members to the group, edit the group file as described in Section 1.2.1.1.

1.3 The Terminal Initialization File

On ULTRIX systems, a terminal special file (device-name) is created for each terminal connected to the system. Terminal special files are listed in the /dev directory. Each special file listed in the /dev directory has an entry in the terminal initialization file, /etc/ttys. The entry contains information used by various routines to initialize and control the use of the terminal. This file can be modified at any time.

Each entry in the terminal initialization file contains five fields separated by spaces or tabs. An entry cannot exceed 512 characters. A field that contains more than one word must be enclosed in quotation marks (""). Unspecified fields default to the empty string or zero, as appropriate. The format of each entry is as follows:

name command type status window="string" description

name This field contains the name of the terminal special file as listed in the /dev directory. All terminals except network pseudoterminals, workstation pseudoterminals, and modems use the following naming convention:

tty[0-9 | A-Z][0-9]

Network pseudoterminals use the following naming convention:

tty[pqrstu][0-9 | a-f]

The modem (dialup) lines have the following convention:

ttyd[0-9 | a-f]

command This field contains the name of the command to be executed each time the terminal is initialized. If the command has arguments, the command and arguments must be enclosed in double quotes.

The command, /etc/getty, is often listed in this field. This command performs such tasks as baud-rate recognition, reading the login name, and calling login(1). This field can also contain other commands such as the startup command for a window system terminal emulator, or a command to maintain other daemon processes.

If the terminal is a pseudodevice, this field should contain the string none.

type

This field contains the type of terminal connected to the terminal special file; for example, a vt100. The possible terminal types are listed in the third field of the /etc/termcap file on your system. If the terminal is a pseudodevice, this field should contain the string **network.**

status

This field specifies the status for each terminal line. When terminals are initialized, these status values are used by the init command. See init(8) in the *ULTRIX Reference Pages*. A terminal line can have up to four status values. They are as follows:

off or **on**
The status bit, on enables logins for the terminal. The status bit, off disables logins for the terminal. If this flag is not set, logins are disabled. For pseudodevices, do not specify this status flag.

modem or **nomodem**
If modem is specified, the terminal line recognizes modem signals (dial-in and dial-out). If nomodem is specified, the terminal line ignores modem signals. The default is nomodem.

secure
If secure is specified, logging in is enabled on this terminal line for root; the flag on must also be set. If secure is not specified, root cannot log in on this terminal line.

shared
If the terminal is shared, the terminal line can be used for both incoming and outgoing connections. If this status field is left blank, the line cannot be used for incoming and outgoing connections.

For example, if this field is **shared**, login, tip, and uucp, can use the same terminal; however, not simultaneously. If this status field remains blank, logins require one terminal, and tip and uucp require a different terminal. See tip(1) and uucp(1c) in the *ULTRIX Reference Pages* for more information.

window="string"

This field is present on ULTRIX Workstations. The *string* contains the name of the X Server for your worksystem. For example:

```
window="/usr/bin/Xqvsm"
```

For more information, see X(1X) and Xqvsm(1X) in the *ULTRIX Reference Pages*.

description This field contains comments. Comments may appear anywhere in the terminal initialization file as long as it is preceded by a number sign (#). The system ignores all comments.

Example 1-1 displays the contents of a terminal initialization file with entries that correspond to the initial configuration. In this example, the initial configuration consists of 8 dmf0 line, 8 dmfl lines, and 32 network pseudoterminal lines.

Example 1-1: Contents of an /etc/ttys File

```
#           "@(#)ttys4.1   (ULTRIX)   1/23/90"
#
#
#
# name    getty      type       status          comments
#
#
#  The dmf0 lines are here:
#
tty00   "/etc/getty 2" vt100     on nomodem secure  # direct connect tty
tty01   "/etc/getty 2" vt100    off nomodem secure  # printer
tty02   "/etc/getty 2" vt100    on nomodem      # direct connect tty
tty03   "/etc/getty 2" vt100    on nomodem      # direct connect tty
tty04   "/etc/getty 2" vt100    on nomodem      # direct connect tty
tty05   "/etc/getty 2" vt100    on nomodem      # direct connect tty
tty06   "/etc/getty 2" vt100    off nomodem          # unused - spare
tty07   "/etc/getty 2" vt100    off nomodem          # unused - spare
#
#  The dmf1 lines are here:
#
tty08   "/etc/getty 2" vt100     off nomodem         # unused
tty09   "/etc/getty 2" vt100     off nomodem         # unused
tty10   "/etc/getty 2" vt100     off nomodem         # unused
tty11   "/etc/getty 2" vt100     off nomodem         # unused
tty12   "/etc/getty 2" vt100     off nomodem              # unused
tty13   "/etc/getty 2" vt100     off nomodem              # unused
tty14   "/etc/getty 2" vt100     off nomodem              # unused
tty15   "/etc/getty 2" vt100     off nomodem              # unused
#
#  The network pty0 pseudodevice lines are here:
#
ttyp0   none         network
ttyp1   none         network
ttyp2   none         network
ttyp3   none         network
ttyp4   none         network
ttyp5   none         network
ttyp6   none         network
ttyp7   none         network
ttyp8   none         network
ttyp9   none         network
ttypa   none         network
ttypb   none         network
ttypc   none         network
ttypd   none         network
ttype   none         network
ttypf   none         network
#
#  The network pty1 pseudodevice lines are here:
#
ttyq0   none         network
ttyq1   none         network
ttyq2   none         network
```

Example 1-1: (continued)

```
ttyq3   none        network
ttyq4   none        network
ttyq5   none        network
ttyq6   none        network
ttyq7   none        network
ttyq8   none        network
ttyq9   none        network
ttyqa   none        network
ttyqb   none        network
ttyqc   none        network
ttyqd   none        network
ttyqe   none        network
ttyqf   none        network
```

For further information on the `/etc/ttys` file, see `getttyent`(3), `gettytab`(5), `termcap`(5), `ttys`(5), and `getty`(8) in the *ULTRIX Reference Pages*.

1.3.1 Modifying the Terminal Initialization File

To modify any field in the terminal initialization file, `/etc/ttys`, use a standard text editor. In most cases, the terminal initialization file is modified for operations such as the following:

- Adding or removing a terminal.
- Enabling or disabling logins to a specific terminal line.
- Enabling or disabling modem recognition (dial in and dial out).
- Setting bit recognition for 7- or 8-bit mode.
- Changing the baud rate for a terminal.

The following sections discuss these topics in more detail. Additionally, Section 3.1.2.6 describes how to process the terminal initialization file after changes have been made.

1.3.1.1 Adding or Removing an Entry – To add a new entry to the terminal initialization file, include the name of the terminal special file, the command that you want executed for that terminal, the type of terminal, and the status of that terminal line. For example:

```
tty01     "/etc/getty std.9600"     vt100     on
```

In the previous example, the terminal special file is named tty01, the `getty` command sets 7-bit recognition at a 9600 baud rate, the terminal type is a VT100, and login is enabled by the the `on` status flag.

To remove a terminal line from the terminal initialization file, edit the file using a standard text editor and delete the entry from the file.

1.3.1.2 Enabling or Disabling Logins – To enable root login on a VT100 terminal, include the following entry:

```
tty02  "/etc/getty std.9600"  vt100 on secure
```

Both the `on` and `secure` status flags must be set to enable login for the root user.

To disable login, set the status bit to `off`.

1.3.1.3 Enabling or Disabling Modem Recognition – To allow modem (dial up) access at 1200 baud on a VT220 without root login privileges, include the following entry:

```
tty01  "/etc/getty std.1200"   vt220 on modem
```

To disable modem access, change the status bit to `nomodem` as follows:

```
tty01  "/etc/getty std.1200"   vt220 on nomodem
```

The baud rate is set by the `getty` command.

1.3.1.4 Setting Bit Recognition – To permit 8-bit recognition at 9600 baud on a VT100 with logins enabled for root, use the following format:

```
tty03  "/etc/getty 8bit.9600" vt100  on secure
```

Note, if a terminal is setup to operate in 8-bit mode and the command field does not specify an 8-bit entry, output to the terminal is displayed as multinational characters.

To permit 7-bit recognition at 9600 baud on a VT100 with logins enabled for root, use the following format:

```
tty03  "/etc/getty std.9600"  vt100  on secure
```

In both examples, 7-bit and 8-bit recognition are set by the `gettytab` entry.

1.3.1.5 Setting the Baud Rate – The baud rate is set by the `getty` command. For example, to set the baud rate to 9600 for a VT220 terminal, use the following format:

```
tty03  "/etc/getty std.9600"  vt220  on secure
```

To change the baud rate, for example, from 9600 to 1200, change the `getty` entry as follows:

```
tty03  "/etc/getty std.1200"  vt220  on secure
```

1.3.1.6 Processing the Terminal Initialization File – Changes made to the terminal initialization file can either be processed during the boot process or during multiuser mode. To process changes during multiuser mode, use the `kill` command as follows:

```
# kill -HUP 1
```

The `kill` command sends a hangup signal to the `init` command which causes `init` to rescan the `/etc/ttys` file. If changes are found in the file, `init` processes those entries that have been modified. For more information, see the `kill`(1) and `init`(8) in the *ULTRIX Reference Pages*.

1.4 The File System Table

The file system table, `/etc/fstab`, contains an entry for each known file system. These entries provide descriptive information on each file system. The order of the entries is important as other programs (such as `dump`, `mount`, and `fsck`) must access this information in sequential order.

Each entry in `/etc/fstab` contains seven fields of information delimited by colons. As defined in the `/usr/include/fstab.h` file, the format of the `/etc/fstab` file is as follows:

spec:file:type:freq:passno:name:opts:

spec This field defines either the file system's block special file (device) name, or it defines a remote file system like Network File System (NFS). For example, `/dev/rz0a` defines a local system and `/usr/src@erie` defines a remote system.

file This field defines the absolute pathname to the directory on which the file system is mounted. The `mount` command uses this information as the default.

type This field specifies the file system mode: `rw` (read-write), `ro` (read only), `rq` (read-write with quotas), `sw` (swap), and `xx` (ignore).

If `sw` is specified and if the file system has been configured for such use, `swapon` (invoked by /etc/rc) makes that file system part of the system swap space. If `xx` is specified, the local file system is ignored, that is, not processed by `mount, dump,` or `fsck.`

freq This field specifies the file system dump frequency (every *n*th day). This is the default order for the `dump` command. For NFS entries, this field should contain a `0` value.

passno This field defines the file system pass number. This is used as the default order for the `fsck` command. Usually, only the root file system has a pass number of 1. The remaining file systems should be assigned higher pass numbers, which enables the `fsck` command to simultaneously check file systems in parallel.

Section 1.4.1.2 discusses this field in more detail.

You should also be aware that NFS entries should have pass numbers of `0` so that local execution of certain commands, such as `fsck` and `dump` ignore these entries. This ensures that you do not interfere with remote file systems maintained by another site.

name This field specifies the type of file system you are mounting. Supported file systems are UFS and NFS.

opts This field defines file system-specific options that are being passed to the file system being mounted.

The following example displays the contents of an `/etc/fstab` file:

```
/dev/ra0a:/:rw:1:1:ufs::
/dev/ra1g:/usr:rw:1:2:ufs::
/usr@bigvax:/bigvax:rw:0:0:nfs::
/usr/uws2.0@bigvax:/usr/uws2.0:rw:0:0:nfs:soft,bg,nosuid:
/usr/dec@bigvax:/usr/dec:rw:0:0:nfs:bg,soft,nosuid:
/usr/pro/xyz@vax:/usr/pro/xyz:rw:0:0:nfs:bg,soft,intr,nosuid:
```

1.4.1 Modifying the File System Table

By convention, the `/etc/fstab` file is created and maintained as a read-only file. Consequently, only the superuser can modify it. Typically, this file is modified for the following reasons:

- To add a new file system
- To remove an obsolete file system
- To change the order in which file systems are loaded, dumped, or checked

- To import a file system with NFS

The following sections discuss these topics in more detail.

1.4.1.1 Adding or Removing a File System – To add a new file system to the `/etc/fstab` file, open the file with a standard text editor. You need the following information to create an entry for the new file system:

- Block special file (device) name
- Absolute pathname to the directory where the file system is located
- Protection mode of the file system
- File system dump number to determine the frequency of dumps
- Pass number supplied for the `fsck` command to determine how the file systems are checked
- The name of the supported file systems (UFS or NFS)

The following example shows an entry for a local file system:

```
/dev/ra0a:/:rw:1:1:ufs::
```

The block special file (device) name is `/dev/ra0a`. The absolute pathname is the `root` directory and the file system is mounted read-write (rw). This file system is dumped daily as signified by the number 1, and it is checked by the `fsck` command on the first pass as indicated by the number 1.

The following example shows an entry for a remote file system:

```
/usr/jwn@minn:/usr/jwn:ro:0:0:nfs:bg,soft:
```

The block special file (device) name is `/usr/jwn@minn`. The absolute pathname is `/usr/jwn` and the file system is mounted read only. The dump field and the pass number fields are defined as zero (0) so that locally executed commands ignore this entry in the `/etc/fstab` file.

To remove a file system entry from the `/etc/fstab` file, open the file with a standard text editor and delete the line. Make certain this does not affect the logical ordering of the other file systems.

1.4.1.2 Changing the Order of File Systems in the /etc/fstab File – The `mount`, `dump`, and `fsck` commands process `/etc/fstab` entries in order, according to the sequential listing of the entries and the pass field. Consequently, the order of the entries in the `/etc/fstab` table is important. This section describes how the `mount`, `dump`, and `fsck` commands use the information in the `/etc/fstab` file.

The `fsck` command performs a file consistency check on local file systems. If the `fsck` command notes any inconsistencies, it attempts to correct the file system before continuing. The `fsck` command makes a number of passes, often inspecting groups of disks in parallel. Hence, all file systems on a single disk should have a different pass number because the `fsck` command can check file systems on the different disks at the same time. To determine which file systems to check in each pass, you must supply a pass number in the `/etc/fstab` file. The root file system should be checked on pass 1; while other root file systems such as partition `a` should be checked on pass 2. Other small file systems can be checked on separate passes. For example, `d` file systems can be checked on pass 3, and `e` file systems can be checked on pass 4. Large user file systems should be checked on the final pass. Those file systems that are NFS mounted, or that you do not want checked should

have a pass number of zero (0). Any file system mounted with read-write (rw) or read only ro are not checked.

The mount command determines which and how to mount local and remote file systems from the entries in the /etc/fstab file. Entries in the /etc/fstab file are read sequentially; therefore, list the file systems as you want them mounted. For example, the mount command fails if it is directed to mount a file system on a directory that has not been mounted. Physically write-protected disks and magnetic tape file systems must be mounted read only (ro) or an error occurs at mount time. Mounting a corrupted file system can cause the system to crash.

In addition to specifying which file systems to mount, you can also specify options when creating /etc/fstab entries for remote file systems. The NFS options direct the mount command to retry a failed mount operation, allows hard mounted file systems to be interrupted, prevents binaries from being executed on a specified file system, and sets the size of the read and write buffers.

The dump command saves a copy of all files changed after a certain date. Using the /etc/fstab file, you can specify how frequently local file systems may be backed up. For example, the number one (1) in the *freq* tells the dump command to back up that particular file system on a daily basis; the number ten (10) tells the dump command to perform a back up every 10 days. For NFS file systems, the *freq* field should contain a zero (0).

See the *ULTRIX Reference Pages* for more information on fsck(8), mount(8), and dump(8).

1.4.1.3 Importing a File System – To mount remote directories or file systems each time your system enters multiuser mode, place an entry in the /etc/fstab file as described in Section 1.4. By placing an entry in the /etc/fstab file, you can automatically mount remote file systems from any NFS server; however, you must also manually create the mount point. The *ULTRIX Guide to System and Network Setup* describes how to use the nfssetup utility to import a file system. For detailed information on importing file systems, see the *ULTRIX Guide to the Network File System.*

1.5 The Aliases File

The aliases file, /usr/lib/aliases, contains information that the sendmail utility uses to route messages to a group of one or more users. Each entry in /usr/lib/aliases contains two fields of information separated by a colon. The format of the /usr/lib/aliases file entries is as follows:

alias:user,user...

alias This field contains the name of the group to which messages are routed.

user This field contains the list of group members (user login names), separated by commas. When an alias is supplied, all user login names included in the entry receive the same mail message. The list of group members can extend beyond one line.

To add comments to the aliases file, include a number sign (#) followed by the comment in first column of the file. The following example shows the contents of an aliases file:

```
# The friends alias is an exclusive group.
friends:barry,candida,martinez
# The team alias lists all project leaders
team:lisa,anthony,terry,alice
# The research alias lists all team members, including project leaders
research:martinez,waker,ellis,artis,wall,lisa,anthony,terry,alice
```

The following sections discuss modifying the `/usr/lib/aliases` file, and how to process new entries in the file.

1.5.1 Modifying the Aliases File

To modify the aliases file, `/usr/lib/aliases`, use a standard text editor. You must edit this file when you want to add new members to a group, remove members from a group, create a new entry, or remove an entry.

To add a new member to a group or create a new entry, open the file using a standard text editor. You can then add the name of a new group member to an existing group, separating the user login name by a comma, or you can create a new entry using the format described in the Section 1.5.

To remove a group member from a group, open the file using a standard text editor, then delete the name and separating comma from the entry. To remove an entire alias, open the file, then delete the entire entry.

1.5.2 Processing the Aliases File

To process the alias file, you can use the `newaliases` command. This command allows new additions to the aliases file to become a part of the `sendmail` aliases database, use the `newaliases` command as follows:

```
# newaliases
```

This command reads the new information added to `/usr/lib/aliases` and rebuilds the `sendmail` aliases database.

For further information, see `newaliases`(1), `aliases`(5), and `sendmail`(8) in the *ULTRIX Reference Pages*

1.6 The Clock Daemon Table

The clock daemon table, `/usr/lib/crontab`, is a symbolically linked file that contains routine commands which the system clock daemon, `cron`, executes at the specified dates and times. For example, `/usr/lib/crontab` might contain routine backup commands as well as commands that cause the automatic removal of outdated or unused temporary files.

Once invoked during multiuser startup, the system activates the system clock daemon, `cron`, every 60 seconds. In turn, the system clock daemon executes those commands listed in the `/usr/lib/crontab` file that are scheduled for that time. Each entry contains six fields of information separated by spaces. The format of the entries in the `/usr/lib/crontab` file is as follows:

minute hour day month weekday command

minute The exact minute that the command sequence is to be executed. The *minute* variable can be 0 through 59.

hour The hour of the day on which the command sequence is to be executed. The *hour* variable can be 0 through 23.

day The day of the month on which the command sequence is to be executed. The *day* variable can be 1 through 31.

month The month of the year on which the command sequence is to be executed. The *month* variable can be 1 through 12.

weekday The day of the week on which the command sequence is to be executed. The *weekday* variable can be an integer from 1 to 7. Monday equals 1 and Sunday equals 7.

command The command sequence that is to be executed. The *command* variable should contain the complete command sequence.

In addition, the first five fields may specify either a single time indicator, a multiple time indicator, a time range, or an asterisk. A single time indicator may consist of one or two consecutive digits such as 3 or 33. A multiple time indicator consists of a string of indicators separated by commas, such as 5,10,15,20. A time range consists of two indicators separated by a dash, such as 5-20. An asterisk field entry represents all times.

The following examples displays the partial contents of a `/usr/lib/crontab` file:

```
# periodic things
0,15,30,45 * * * * (echo '^M' `date`; echo '') >/dev/console
0,15,30,45 * * * * /usr/lib/atrun

# daily stuff
5 4 * * * sh /usr/adm/newsyslog
15 4 * * * ( cd /usr/preserve; find . -mtime +7 -a -exec rm -f {} ; )
20 4 * * * find /usr/msgs -mtime +21 -a ! -perm 444 -a ! -name bounds
     -a -exec rm -f {} ;

# NOTE: The above line is wrapped.

# local cleanups
30 4 * * * find /usr/spool/mqueue -type f -mtime +5 -name df -exec rm {} ;
35 4 * * * find /usr/spool/mqueue -type f -mtime +5 -name tf -exec rm {} ;
40 4 * * * find /usr/spool/rwho -type f -mtime +21 -exec rm {} ;
#
```

The next two sections discuss the `cron` command and describes how to modify the `/etc/usr/crontab` file.

1.6.1 Specifying cron

The `cron` command executes at specified dates and times according to the information in the `/usr/lib/crontab` file. The `cron` command never exits; hence, it should only be executed once to avoid using up system resources. For the best results, you should run `cron` command from the initialization process by including it in the `/etc/rc` file. For more information, see the `init`(8) command in the *ULTRIX Reference Pages*.

1.6.2 Modifying the Clock Daemon Table

Each entry in /usr/lib/crontab contains information that specifies a time and command sequence that is to be executed regularly, hence, stagger the crontab entry times so that the processes are not running at together. When appropriate, and especially during anticipated periods of heavy user activity, include the nice command in your crontab file entries. The nice command tells cron to execute commands at a lower priority.

To change the clock daemon table, use a standard text editor to open and edit /usr/lib/crontab. For further information on the clock daemon and prioritizing tasks, see cron(8) and nice(1) in the *ULTRIX Reference Pages*.

1.7 The Message-of-the-Day File

The message-of-the-day file, /etc/motd, provides system users with information relevant to each day's operation. The message-of-the-day file is displayed on the terminal screen after each login.

The /etc/motd file does not have a special format; however, the first line of the message-of-the-day file is controlled by the /etc/rc.local file. This line contains the adjective ULTRIX, the current version of the operating system, revision number, and the day's date. To modify the motd file, use a standard text editor; you must have root privileges.

The system displays the same message after each login until you either modify or delete the contents of /etc/motd. The following example displays the contents of an /etc/motd file:

```
Ultrix V4.0 (Rev 162) System #8: Fri May 11  10:45:34 EST 1990

-------------------------------------------------------------------------

                This machine is running ULTRIX Version 4.0

-------------------------------------------------------------------------

On Monday, May 15, 1990, this machine will be unavailable
from 7am to 10am.  Field Service is performing some maintenance
tests.

-------------------------------------------------------------------------

If you encounter problems or have questions
     concerning this machine, send mail to the
     admin account.
```

Adding Devices 2

This chapter describes how to add terminals, pseudoterminals, and device drivers to your system. Before you add a new device to your system, make certain that the configuration file contains a description of that device type. The system configuration files reside in the following directories:

- For VAX processors, the system configuration file resides in the `/usr/sys/conf/vax` directory.
- For RISC processors, the system configuration file resides in the `/usr/sys/conf/mips` directory.

If the device description is in the configuration file, use the `MAKEDEV` command as described in this chapter. However, if the device description is not in the configuration file, you must rebuild and boot a new kernel as described in the *ULTRIX Guide to Configuration File Maintenance* before making the devices known to the system.

When you add a new device to your system configuration with the `MAKEDEV` command, a special file is created for the device in the `/dev` directory. Use the `file` command to display information about the special files in the `/dev` directory as follows:

```
% file /dev/* | more
```

For reference information about making special devices and building a new kernel, see `MAKEDEV`(8) and `doconfig`(8) in the *ULTRIX Reference Pages*.

2.1 Adding Terminals and Pseudoterminals

Before you can add terminals and pseudoterminals to your system, you must physically connect the devices to your system, then make the devices known to the system.

The following sections describe how to physically connect the devices to your system and how to make the devices known to your system.

2.1.1 Physically Connecting Terminal Lines and Multiplexers

To physically connect terminal lines and multiplexers to your system, use the following steps:

1. Log on as root.
2. Stop system activity by using the `/etc/shutdown` command and then power down the system by turning off the processor or by issuing the correct console command. For further information, see the *ULTRIX Guide to Shutdown and Startup*.

3. If appropriate for your device, set the control status registers (CSRs) and interrupt vectors on the board. Most Digital devices have standard addresses. Refer to the appropriate device manual for the standard address.
4. Install the board.
5. Power up the machine.
6. Boot the system. For information on how to boot your particular processor, refer to the *ULTRIX Guide to Shutdown and Startup*.

2.1.2 Adding Terminals

After physically connecting the terminal line or multiplexer, use the MAKEDEV command to logically add the terminal to your system as follows:

1. Use the cd command to move to the /dev directory.

```
# cd /dev
```

2. Create the device special files using the MAKEDEV command.

MAKEDEV *device#*

The *device* variable is the device mnemonic for the terminal multiplexer you are adding. Appendix A lists each device and its mnemonic. The number sign (#) represents the number of device (0 through 10).

For example, to create a device file for a UNIBUS DMZ32 comm multiplexer, unit 3, type the following command:

```
# MAKEDEV dmz3 > terminal.file
```

MAKEDEV responds with a listing of the special device files it has created for this device. For example, if the first 20 special device files (tty00 – tty19) are allocated, the display appears as follows:

```
MAKEDEV: special file(s) for dmz3:
tty20 tty21 tty22 tty23 tty24 tty25 tty26 tty27 tty28 tty29
tty30 tty31 tty32 tty33 tty34 tty35 tty36 tty37 tty38 tty39
tty40 tty41 tty42 tty43
```

Save the output generated by the MAKEDEV command. You need this information to update the /etc/ttys file.

3. Update the /etc/ttys file by including the new tty lines. Each entry corresponds to the special device file created by the MAKEDEV command.

For example, using the sample output shown in Step 2, to add an entry for the new tty20 line, enter the following in the /etc/ttys file:

```
tty20 "etc/getty std.9600" vt100 on nomodem  #direct connect tty
```

For a detailed description of the /etc/ttys file, see Section 1.3.

4. Add the device definition to the system configuration file if it is not present. After adding the device definition, rebuild and boot the kernel. For information on editing the system configuration file, see the *Guide to Configuration File Maintenance*.

2.1.3 Adding Pseudoterminals

To add a pseudoterminal to your system, follow these steps:

1. Use the `cd` command to move to the `/dev` directory.

   ```
   # cd /dev
   ```

2. Create the special device files using the `MAKEDEV` command.

   ```
   MAKEDEV pty#
   ```

   The number sign (#) represents the set (0 through 10) of pseudoterminals you want to create. Each single set creates 16 pseudoterminals, for a total of 176 in the system.

   Note

   By default, the installation software creates special device files for the first two sets of pseudoterminals, `pty0` and `pty1`. The `pty0` pseudoterminals have corresponding special device files named `/dev/ttyp0` through `/dev/ttypf`. The pty1 pseudoterminals have corresponding special device files named `/dev/ttyq0` through `/dev/ttyqf`.

   If you add pseudoterminals to your system, the `pty#` variable must be than `pty1` because the installation software creates sets `pty0` and `pty1`. For example, to create special device files for a third set of pseudoterminals, type the following:

   ```
   # MAKEDEV pty2 > pseudo.file
   ```

   `MAKEDEV` responds with a listing of the special device files it has created.

   Save the output generated by the `MAKEDEV` command. You need this information to update the `/etc/ttys` file. For a detailed description of the `/etc/ttys` file, see Section 1.3.

3. Add the new line entries to the `/etc/ttys` file. The `/etc/ttys` file entries correspond in name to the special device files that `MAKEDEV` created.

   For example, if you are adding a third set of pseudoterminals to your system, `MAKEDEV` creates the device special files `/dev/ttyr0` through `/dev/ttyrf`. You can then edit the `/etc/ttys` file to include lines `ttyr0` through `ttyrf`.

4. Edit the pseudo-device `pty` entry in the system configuration file.

   By default, the kernel supports 32 pseudoterminals. The configuration file's pseudo-device `pty` entry for the default case is `pseudo-device pty`. If you are adding more pseudoterminals to your system, you must edit this system configuration file entry.

   For example, to add another set of 16 pseudoterminals to your system, edited the configuration file entry and increment the number 32 by 16 as follows:

   ```
   pseudo-device pty 48.
   ```

   For more information on the configuration file and its pseudo-device line entry, see the *ULTRIX Guide to Configuration File Maintenance*.

5. Rebuild and boot a new kernel.

2.2 Adding Disk and Tape Drives

Before you can add new tape or disk drives to your system, you must install the new controller at a Digital standard address, run diagnostics, physically connect the devices, and then make the devices known to the system.

The following sections describe how to physically connect the devices to your system and how to make the devices known to your system.

2.2.1 Physically Connecting the Disk and Tape Devices

Use the following steps to physically connect the tape and disk devices to your system:

1. Stop system activity by using the `/etc/shutdown` command and then power down the system by turning off the processor or by issuing the correct console command. For further information, see the *ULTRIX Guide to Shutdown and Startup*.
2. If appropriate for your device, set the control status registers (CSRs) and interrupt vectors on the board.
3. Install the board.
4. Power up the machine.
5. Boot the system. Refer to the *ULTRIX Guide to Shutdown and Startup* for information on booting your processor.

2.2.2 Adding Both Disk and Tape Devices

Once you have physically connected the tape or disk drive, you must make the tape or disk drive know to your system. Use the following steps to add a new tape or disk drive:

1. Use the `cd` command to move to the `/dev` directory:

   ```
   # cd /dev
   ```

2. Invoke the `MAKEDEV` command using the following syntax:

   `MAKEDEV` *device#*

   The *device* variable is the device mnemonic for the drive you are adding. Appendix A lists the device mnemonics for all supported disk and tape drives. The number sign (#) is the number of the device, 0 through 31 for tape, 0 through 95 for MSCP disks.

   For example, to create the device special files for two SCSI disk drives, type the following command:

   ```
   # MAKEDEV rz0 rz1
   ```

3. If a device definition for the additional drive does not currently exist in your system configuration file, you must edit the file and include the new definition, then you must build and boot a new kernel. For information on configuration file device specifications, see the *ULTRIX Guide to Configuration File Maintenance*.

2.3 Setting Up the System Console

The default console entry in the /etc/ttys file is set to e for the baud rate as follows:

```
console "/etc/getty e"  dw3     on secure      # console terminal
```

If you have a hard-copy console, you can accept this default. However, if you have a CRT console, you must change the entry in the /etc/ttys file to std.9600. For example:

```
console "etc/getty std.9600"  vt100  on secure   # console terminal
```

Managing the Print System 3

The ULTRIX operating system supports a variety of printers that can be accessed on the local or remote network. This chapter discusses how to manually set up and modify the print system, and additionally describes the commands available to maintain the print system. Ensure that you have installed the appropriate print software subset before setting up the printer. On VAX machines, install the ULTPRINT400 subset; on RISC machines, install the UDTPRINT400 subset.

If you are not familiar with the ULTRIX print system, use the `lprsetup` facility to set up and modify the print system. The `lprsetup` facility is an interactive facility that steps you through the set up of the print system. For instructions on using the `lprsetup` facility, see the *ULTRIX Guide to System and Network Setup*.

3.1 Setting Up the Print System Manually

To set up or modify the print system manually, you must complete the following:

- Create a device special file (device-name) for each printer using the `MAKEDEV` command.
- Update the terminal initialization file, `/etc/ttys`, if a printer is connected to the system by a serial line.
- Update the printer capability database, `/etc/printcap`.

The following sections discuss these operations in detail.

3.1.1 Creating a Device Special File

For each printer connected to your system, you must create a device special file in the `/dev` directory. By default, a device special file may have been created for some printers during the installation process; however, most device special files must be created using the `MAKEDEV` command. A device special file name is specified as follows:

- `/dev/lp`*n* for printers attached by parallel interfaces
- `/dev/tty`*nn* for printers attached by all serial interfaces

The *n* arguments specify the port (physical connection) of each local printer connected to your system. To determine the port number of a printer, refer to the *Site Management Guide* prepared by the Digital field service person who installed the printer.

To create a device special file in the `/dev` directory for each printer, set your default directory to the `/dev` directory, then type the `ls` command to determine which device special files exist. For example, to determine if a device special file exists for `lp1`, type the following commands:

```
# cd /dev
# ls -l lp1
```

If the file exists, and the printer is attached by a parallel interface, you need only define this entry in the /etc/printcap file. If the file exists, and the printer is attached by a serial interface, you must edit the /etc/ttys file, then define the entry in the /etc/printcap file.

If the file does not exist, use the MAKEDEV command to create the device special file. For example, to create a device special file for lp1, type the following:

```
# MAKEDEV lp1
```

Section 3.1.2 provides more information on the /etc/ttys file. Section 3.1.3 describes the /etc/printcap file.

3.1.2 Modifying the Terminal Initialization File

The terminal initialization file, /etc/ttys, must be modified if you have a serial line printer. The device special file names for these printers use the same format for file names as the terminal special files described in the /etc/ttys file. Hence, use a standard text editor to edit the /etc/ttys file to disable terminal capabilities for those entries that correspond to serial printer device specifications. To disable terminal capabilities in the /etc/ttys file, the *status* flag must specify **off**. See Section 1.3 for detailed information on editing the /etc/ttys file.

3.1.3 Updating the Printer Capability Database

The printer capability database, /etc/printcap, contains a descriptive entry for each printer available on your system. A generic version of the /etc/printcap file is created during the installation process; however, you must edit this file to define each printer's capabilities.

Entries in the /etc/printcap file consist of several fields separated by colons. An entry can span several lines; however, a backslash (\), used as a line continuation character, must appear at the end of each entry to signify that the line is continued. The first line of an entry usually specifies the printer's logical names. Subsequent lines define the capabilities of each printer. If you do not specify a capability, the default is used. The following example shows a printcap entry:

```
lp2|2|lab2|lg01 printer in lab:\          1
        :lp=/dev/lp2              2
        :rm=atlanta                       3
        :rp=lp2                   4
        :sd=/usr/spool/lp2                      5
        :mx#0:                    6
```

In the previous example, the printcap entries are defined as follows:

1 The first line specifies the logical name (lp) and number (2) of the printer as listed in the /dev directory. This line also contains any other logical names (lab2) for the printer and lists the location of the printer.

2 The **lp** symbol specifies the name of the special file to open for output.

3 The **rm** symbol specifies the machine name of the remote printer as this printer is not connected to the local system.

4 The **rp** symbol specifies the logical name for the printer on the remote system.

5 The **sd** symbol specifies the spooling directory where print requests are queued before printing.

6 The **mx** symbol specifies the maximum allowable file size in blocks. In this instance, zero (0) removes any file size restriction.

The ULTRIX operating system provides you with a file, `/etc/printcap.examples`, which contains examples of printer definitions. By using this file as a template, you can tailor it to suit the needs of your site. To display this file, type the following:

```
# more /etc/printcap.examples
```

Table 3-1 lists and describes the available printcap symbols and their defaults if applicable. The format for specifying a printcap entry is as follows:

symbolname=value

Table 3-1: Printcap Symbols

Printcap Symbols	Description
af	Specifies the accounting file that tracks the number of pages printed by each user for each printer. Each accounting file associated with a printer must have a unique name. This file must be owned by the print daemon. This symbol cannot be used for remote printer entries. When this symbol is specified, intermediate directories are automatically created as needed.
br	Specifies the baud rate for the printer. This symbol must be specified with tty devices (serial lines); however, it has no effect on printers connected to the console port or printers connected by a parallel port.
cf	Specifies the output filter for the cifplot data filter. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages.*
ct	Specifies the connection type. Valid connection types are *dev*, *lat*, *remote*, and *network*. The *dev* argument specifies a printer connected to the local system. The *lat* argument specifies a printer connected over the local area. The *remote* argument specifies a printer connected to another system running a compatible printer daemon. The *network* argument specifies a printer that has does not use standard output, has its own cpu, address, and node name.
Da	Specifies the default data type used by the print daemon. Valid arguments are *ansi, ascii, postscript, regis*, and *tek*. If a data type other than *postscript* is specified, a translator is invoked by the daemon to convert the files in the job to *postscript*. This symbol is only valid with PostScript (TM) printers.

Table 3-1: (continued)

Printcap Symbols	Description
df	Specifies the output filter for the TeX data filter (DVI format). For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*.
Dl	Specifies the device control module library file. The default is `/usr/lib/lpdfilters/lps_v3.a`. This symbol is only valid with PostScript (TM) printers.
dn	Specifies the name of the `daemon` program that must be invoked when a print request is made of the printer. The default is `/usr/lib/lpd` which should not be changed. This symbol enables you to specify support for line printer daemons other than the default.
du	Specifies the daemon UID used by the print spooler programs. The default value is zero (0) which should not be changed. This symbol enables you to define other daemon UIDs for printer daemons other than `/usr/lib/lpd`.
fc	Specifies which terminal flag bits to clean when initializing the printer line. All bits should be cleared (fc=0177777) before calling the **fs** symbol. For more information, see the **fs** symbol in this table.
ff	Specifies the string to send as a form feed to the printer. The default is `\f`.
fo	Specifies whether a form feed is printed when the device is first opened. This is in addition to the normal form feed which is printed by the driver when the device is opened. To suppress all printer induced form feeds, specify this symbol with the **sf** symbol.
fs	Specifies which terminal flag bits to set when initializing the printer line. Before calling the **fs** symbol, all bits should be cleared using the **fc** symbol, then the **fs** symbol should be set to specific bits. For detailed information on these bits, see `tty`(4) in the *ULTRIX Reference Pages*. A brief description of each bit follows:

Flag	Value	Description
ALLDELAY	0177400	Delay algorithm selection
BSDELAY	0100000	Select Backspace delays (not implemented)
BS0	0	
BS1	0100000	
VTDELAY	0040000	Select form feed and vertical tab delays
FF0	0	

Table 3-1: (continued)

Printcap Symbols	Description		
	ff1	0100000	
	CRDELAY	0030000	Select carriage return delay
	CR0	0	
	CR1	0010000	
	CR2	0020000	
	CR3	0030000	
	TBDELAY	0060000	Select tab delays
	TAB0	0	
	TAB1	0002000	
	TAB2	0004000	
	XTABS	0006000	
	NLDELAY	0001400	Select new-lines delay
	NL0	0	
	NL1	00E00400	
	NL2	0001000	
	NL3	0001400	
	EVENP	0000200	Even parity allowed on input (most terminals)
	ODDP	0000100	Odd parity allowed on input
	RAW	0000040	Raw mode: wake up on all characters; 8–bit interface
	CRMOD	0000020	Map CR into LF; echo LF or CR as CR-LF
	ECHO	0000010	Echo (full duplex)
	LCASE	0000004	Map upper case to lower on input
	CBREAK	000000	Return each character as soon as it is typed
	TANDEM	0000001	Automatic flow control

gf — Specifies the graph data filter (`plot`(3X) format). For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*.

ic — Specifies the driver that supports nonstandard ioctl to an independent printout.

Printcap Symbols	Description
if	Specifies the accounting filter. If an accounting filter is specified using the **af** symbol, the **if** symbol is ignored. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*. Available print filters are listed below. For detailed information on the individual filters, see Section 8 of the *ULTRIX Reference Pages*:

Filter name	Description
`/usr/lib/lpdfilters/lpf`	Line printer filter (LP25, LP26, LP27, LP29, LG01, LA210, LQP02, LQP03)
`/usr/lib/lpdfilters/lqf`	Letter quality filter (LQP02, LQP03)
`/usr/lib/lpdfilters/ln01of`	LN01 Laser Printer filter
`/usr/lib/lpdfilters/ln03of`	LN03 Laser Printer filter
`/usr/lib/lpdfilters/ln03of`	LN03S Laser Printer filter
`/usr/lib/lpdfilters/lcg01of`	LCG01 Color Printer
`/usr/lib/lpdfilters/lj250of`	LJ250 DEColorwriter filter

It	Specifies the default input tray. Valid arguments are *bottom*, *middle, lcit*, and *top*. An error occurs if the specified input tray is not available on the printer. This symbol is only valid with Postscript (TM) Printers.
lf	Specifies the name of the error log file. The default is `/dev/console`. If more than one printer is connected to your system, each printer must have an error log file with a unique name. When the error log file is created, intermediate directories are automatically created as needed.
Lf	Specifies the Layup to PostScript (TM) translator program. The default is `/usr/lib/lpdfilters/layup`. This symbol is only valid with PostScript (TM) printers.
Lu	Specifies the layup definition file which contains information that can alter the appearance of output such as margins and borders. This symbol is only valid with PostScript (TM) printers.
lo	Specifies the name of the lock file used by the printer daemon to control the printing jobs in each spooling directory. The default is lock. This symbol enables you to define a lock file for use by other printer daemons.

Printcap Symbols	Description
lp	Specifies the name of the special file to open for output. The default is `/dev/lp` which specifies a parallel printer. Valid special file names include lp1, lp2, lp3, and so on. Serial printers should contain a special file name such as tty0, tty1, tty2, and so on. You must define this entry with a null argument for remote printers.
mc	Specifies the maximum number of copies that may be printed using the `lpr` command. See `lpr`(1) in the *Ultrix Reference Pages.*
Ml	Specifies the action to take with user errors detected by the printserver. Valid arguments are *keep* and *ignore*. If you specify *keep*, the messages are recorded in a file and sent to you. If you specify *ignore*, the messages are not recorded. This symbol is only valid with PostScript (TM) printers.
mx	Specifies the maximum allowable file size (in BUFSIZE blocks) printable by each user. If this symbol is not specified, 1000 blocks is the maximum allowable file size. If you specify zero (mx=0), the file size restriction is removed.
nc	Does not allow control characters to appear in the output file.
nf	Specifies a ditroff filter. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages.*
Nu	Specifies the default number of pages on a single sheet. The argument can be a number from 1 through 100. This symbol is only valid with PostScript (TM) printers.
of	Specifies the output filter to be used with the printer. Output filters filter text data to the printer device when accounting is not enabled or when text data must be passed through a filter. If the **af** symbol is specified, the **of** symbol is ignored. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages.* Available output filters are listed below. For detailed information on the following filters, see Section 8 of the *ULTRIX Reference Pages.*

Filter Name	Description
`/usr/lib/lpdfilters/lpf`	Line printer filter (LP25, LP26, LP27, LP29, LG01, LA210, LQP02, LQP03)
`/usr/lib/lpdfilters/lqf`	Letter quality filter (LQP02, LQP03)
`/usr/li/lpdfilters/la75of`	LA75 Dot Matrix Printer filter

Printcap Symbols	Description
	/usr/lib/lpdfilters/ln01of LN01 Laser Printer filter /usr/lib/lpdfilters/ln03of LN03 Laser Printer filter /usr/lib/lpdfilters/ln03of LN03S Laser Printer filter /usr/lib/lpdfilters/lcg01of LCG01 Ink Jet Printer filter /usr/lib/lpdfilters/lg02of LG02 Ink Jet Printer filter /usr/lib/lpdfilters/lg31of LG31 Line Printer filter /usr/lib/lpdfilters/lj250of LJ250 Ink Jet Printer filter
op	Specifies the object port on a LAT terminal server.
Or	Specifies the manner by which pages are printed. Valid arguments are *portrait* and *landscape*. This symbol is only valid with PostScript (TM) printers.
os	Specifies the object service on a LAT server. (Not Used)
Ot	Specifies the output tray. Valid arguments are *face-up, lcos, lower, side, top*, and *top*. If the specified output tray is not available on the printer, an error occurs. This symbol is only valid with PostScript (TM) printers.
pl	Specifies the page length in lines. The default is 66 lines.
pp	Specifies the print command filter replacement. The available filter is `/usr/lib/lpdfilters/ln01pp` which is the LN01 laser printer filter. This filter replaces the **pr** filter request that is made when using the `lpr` command with the **–p** option. See `lpr`(1) in the *ULTRIX Reference Pages*. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*.
ps	Specifies the mode in which the daemon runs. Valid arguments are *non_PS* (non-postscript printers) and *LPS* (supported postscript printers).
Ps	Specifies the page size. Valid arguments are *a, letter, a3, a4, a5, b, ledger, executive*, and *legal*. If the specified page size is not available on the printer, an error occurs. This symbol is only valid with PostScript (TM) printers.
pw	Specifies the page width in characters. The default page width is 132 characters; however, you should not specify more than 80 characters for a letter quality printer that uses 8 1/2 by 11 paper.

Printcap Symbols	Description
px	Specifies the page width in pixels.
py	Specifies the page length in pixels.
rf	Specifies the filter for printing FORTRAN style text files.
rm	Specifies the machine name for a remote printer. This symbol may only be used in printcap entries for remote printers. In addition to this symbol, you must also define the printcap symbols **lp**, **rp**, and **sd** to access a printer on the remote system. A remote printer cannot process a print request if the hostname of the requesting node does not appear in the `/etc/hosts.lpd` or `/etc/hosts.equiv` files of the local and remote systems. Note that an asterisk (*) at the start of any line in the `/etc/hosts.lpd` enables remote print requests from all systems.
rp	Specifies the remote print name argument. The name specified must be one of the logical names for the printer on the remote system. This symbol must be specified for a remote printer.
rs	Restricts the remote printer usage to those users with local accounts.
rw	Specifies that the printer has read and write access. Usually, a printer allows only write access.
sb	Specifies a one-line banner.
sc	Suppresses multipile copies. This is equivalent to setting the **mc** symbol to one (1).
sd	Specifies the spooling directory where all print requests are queued before they are printed. The default is `/usr/spool/lpd`. Each printer connected to the system should have a unique name. Both local and remote printcap entries must specify a spooling directory. If you use a directory other than the default, you must create the directory. When the spooling directory is created, the intermediate directories are automatically created as needed.
Sd	Specifies the default sheet size value. Valid arguments are *a, letter, a3, a4, a5, b, ledger, b4, b5, executive*, and *legal*. Unlike the **Ss** symbol, if a sheet size is specified that is not available, an error does not occur and the print job is printed on the paper available with the printer. If the **Ss** symbol is specified, the **Sd** symbol is ignored. This symbol is only valid with PostScript (TM) printers.

Printcap Symbols	Description
sf	Suppresses all printer induced form feeds, except those present in the file. The **sf** symbol, when used with the **sh** symbol, is useful if you want to print a letter on a single sheet of stationary.
sh	Suppresses printing of the burst page header. See the **sf** symbol for more information.
Si	Specifies the default sides option. Valid arguments are *one_sided_duplex, 2, two_sided_duplex, tumble, two_sided_tumble, one_sided_duplex, one_sided_tumble,* or *two_sided_simplex.* An error occurs if the specified argument is not available on the printer. For a description of these arguments, see the *–K* argument to `lpr`(1) in the *ULTRIX Reference Pages.* This symbol is only valid with PostScript (TM) printers.
Ss	Specifies the physical sheet size. Valid arguments are *a, letter, a3, a4, a5, b, ledger, b4, b5, executive,* and *legal.* An error occurs if the specified sheet size is not availabe on the printer. This symbol overrides the **Sd** symbol. This symbol is only valid with PostScript (TM) printers.
st	Specifies the status file name. The default name is **status**. The status file is located in the spooling directory. The status of the printer is written to this file.
tf	Specifies a troff data filter. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages.*
tr	Specifies a trailing string to print when the spooling queue empties. This symbol resets the printer to a known state. The trailing string may be a series of form feeds or escape sequence.
ts	Specifies a LAT terminal server node name.
uv	Specifies the ULTRIX version. This symbol is used by the daemon to determine whether to expand percent (%) escapes when using filter capabilities. Valid arguments are **3.0** and **4.0**. When using the **ct** symbol, you must use the **uv** symbol. For more information on the symbols **ct** and **uv**, see `lpd`(8) in the *ULTRIX Reference Pages.*
Ul	Specifies the default upper page limit value. This must be a value from 1 through 10,000. This symbol is only valid with PostScript (TM) printers.

Printcap Symbols	Description
vf	Specifies a raster image filter. The available raster filter that can be specified with is `/usr/lib/lpdfilters/ln01vf` which is the LN01 laser printer filter. Other raster filters can be specified with the **if** or **of** symbols. For detailed descriptions of filters, see Section 8 of the *ULTRIX Reference Pages*. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*.
xc	Specifies the local mode bits to clear when the terminal line is first opened. You should clear all bits by specifying xc=0177777 before specifying the **xs** symbol. See the description of the **xs** symbol for more information.
xf	Specifies the pass-thru filter name. This routine is specified when output is preformatted and does not require special filtering. For more information on using filter capabilities, see `lpd`(8) in the *ULTRIX Reference Pages*.
Xf	Specifies the translator dispatch program. The default is `xlator_call`. This symbol is only valid with PostScript (TM) printers.
xs	Specifies the local mode bits to set when the terminal line is first opened. You should clear all bits by specifying the **xc** symbol before specifying the **xs** symbol. For a detailed description of the status bits, see `tty`(4) in the *ULTRIX Reference Pages*. A brief description of the status bits follow:

Flag	Value	Description
LCRTBS	0000001	Backspace on erase rather than echoing erase
LPRTERA	0000002	Printing terminal erase mode
LCRTERA	0000004	Erase character echoes as backspace-space-backspace
LTILDE	0000010	Convert a tilde (~) to an apostrophe (') on output (for Hazeltine terminals)
LLITOUT	0000040	Suppress output translations for 8–bit
LTOSTOP	0000100	Send SIGTOC for background output
LFLUSHO	0000200	Output is being flushed
LNOHANG	0000400	Do not send hangup when carrier drops
LAUTOFLOW	0001000	Hardware responds to flow contro characters. (See Flow control.)
LCRTKIL	0002000	BS-space-BS erase entire line

Printcap Symbols	Description		
			on line kill
	LPASS8	0004000	Allow 8–bit characters in input and output
	LCTLECH	0010000	Echo input control characters as CTRL/X; delete as CTRL/?
	LPENDIN	0020000	Retype pending input at next read or input character
	LDECCTQ	0040000	Only CTRL/Q restores output after CTRL/S, like DEC systems
	LNOFLSH	0100000	Flush output on receipt of suspend character

3.2 Controlling Print Jobs

Once known to your system, the print system software requires little maintenance. This section describes the commands and files you use to maintain the print system. They are as follows:

`/usr/lib/lpd` This program, the line printer daemon, is a print spool handler. Normally, the program is invoked at boot time from the `rc` file. The daemon works with several system programs and files to coordinate and synchronize printer activity. The ULTRIX operating system supplies this file. While you do not modify the file, you can specify spooling, logging, and locking activities. You must have superuser privileges to access this program.

`/etc/lpc` This program lets you control the operation of the line printer system. While most control functions are available only to the superuser, some functions can be accessed by general users.

`/usr/ucb/lpr` This program lets you queue and submit files for printing.

`/usr/ucb/lpq` This program lets you examine the status of jobs currently on the print queue.

`/usr/ucb/lprm` This program lets you remove jobs from the print queue.

`/etc/pac` This program generates accounting information about printer use at your site. You must have superuser privileges to use this program.

The following sections describe how to use these files and commands. For more information on these commands, see the *ULTRIX Reference Pages*.

3.2.1 The Line Printer Daemon

The line printer daemon, `lpd`, provides network communications of print requests. It also provides the selection and start of specific print filters for specific print requests. The print filters process the varying input formats into printer-specific output format.

The line printer daemon interface is a task that runs automatically and remains running, ready for input. This daemon is generally started at boot time from the `/etc/rc` file. The `lpd` command invokes the line printer daemon. When you submit a print job using the `lpr` command, the line printer daemon schedules jobs and notifies printers that have jobs waiting.

When signaled for inputs, the line printer daemon checks the spooler directory, `/usr/spool/lpd`, for the existence of a lock file. If the lock file exists, `lpd` knows another job is currently printing. If a lock file is not present, `lpd` creates one to reserve access to the printer for a particular print job. Once the daemon creates the lock file, it scans the directory of files beginning with `cf`. These files are control files which represent print jobs. For example:

```
% lpr memo.1
% ls -l /usr/spool/lpd
total 3
-rw-rw----    1 daemon         86 July 9 11:11 cfA024myvax
-rw-rw----    1 dmf          2358 July 9 11:11 dfA024myvax
-rw-r--r--    1 root            5 July 9 11:11 lock
-rw-rw-r--    1 root           52 July 9 11:11 status
```

The control file beginning with `cf` contains print instructions and the data file beginning with `df` contains the formatted text. The lock file contains the process ID of the currently running daemon, while the status file contains a line describing the current printer status.

3.2.2 Controlling Printer Activity

The `lpc` command enables you to control the activity of the line printers and spooler queues listed in `/etc/printcap`. Use the `lpc` command to do the following:

- Enable/disable a printer
- Enable/disable a spooler queue
- Alter order of queued jobs
- Display printer, queue, or daemon status

You must have superuser privileges to enable or disable a printer or queue, or to alter the order of queued jobs.

3.2.3 Printing a File

The `lpr` command queues and submits a job for printing. For example, if you have a file named memo.1, you can print the file using:

```
% lpr memo.1
```

The `print` command paginates the job before printing. To do this type the following:

```
% print memo.1
```

If you pipe the file to `lpr`, the file name is listed as standard input. For example:

```
% cat memo.1 | lpr
```

If a file is not specified, the standard input is read.

3.2.4 Checking the Print Queue

The `lpq` command displays the current contents of the line printer queue and lists the jobs that have not yet printed. For example:

```
% lpq
lp is ready and printing
Rank      Owner    Job   Files                 Total Size
active    dmf      24    memo.1                23056 bytes
1st       dmf      25    (standard input)       6987 bytes
```

There are two jobs in the print queue belonging to user `dmf`. The active job is number 24, memo.1. The `lpq` command displays information in the order in which it is scheduled to print.

3.2.5 Removing a Job from the Queue

The `lprm` command allows you to remove a job from a queue. To locate the job number and then remove a print job, type:

```
% lpq
lp is ready and printing
Rank   Owner   Job   Files             Total Size
active dmf     24    memo.1            23056 bytes
1st    dmf     25    /etc/printcap      6987 bytes
% lprm 24
dfA024myvax dequeued
cfA024myvax dequeued
```

When used without arguments `lprm` deletes the currently active job, if it is owned by you or if you are the superuser. If invoked with a user's name, it removes all print jobs owned by that user.

3.2.6 Generating a Report of Printer Use

The `pac` command displays a report detailing number of pages printed, feet of paper consumed, and total estimated cost per user. To periodically generate a report of your printer activity, type the following:

```
# /etc/pac
```

Note that the `pac` command can be used only if you have specified an accounting file for each printer for which a report is wanted.

Adding and Deleting Software Subsets 4

The `setld` command is used to install and manage software. You can save disk space with the `setld` command by specifying and loading only the software subsets that you need. This chapter describes how to use the `setld` command. You must have superuser privileges to modify the system software using the `setld` command. The `setld` command has several command line syntaxes to list, load, add, delete, and verify software subsets. Although the command lines differ, the `setld` command lines use certain command options as follows:

- The *dir* argument specifies the destination of the subset. Specify this argument if you are loading the software to a file hierarchy that starts somewhere other than at `root` (*/*).
- The optional *subset* argument specifies the name of the subset.
- The *location* argument specifies the device special file or mount point containing the media from which the subset or product is to be transferred.

If you specify a *dir* argument, it must precede the command option. If you specify a *subset* or *location* argument, it must follow the command option.

For detailed information on the `setld` command, its options, and the command line syntax, see `setld`(8) in the *ULTRIX Reference Pages*.

4.1 Listing Software Subsets

To display the status of all subsets known to the system, specify the `setld` command with the `-i` (inventory) option. The format is as follows:

`setld [`**–D** *dir*`] -i [`*subset*`]`

For example, to display a list of subsets, type the following:

```
# setld -i
```

To display the files included in a particular subset, for example, UDTUUCP400, type the following:

```
# setld -i UDTUUCP400
```

4.2 Loading Subsets

To load a software product to your system for the first time, use the `setld` command with the `-l` (load) option. The format is as follows:

`setld [`**–D** *dir*`] -l <`*location*`>`

This tells the `setld` program to load the mandatory subsets and the optional subsets selected during the installation.

For example, to load the UDTUUCP400 subset to an offline system rooted at `/mnt` from tape unit 2, type the following:

```
# setld -D /mnt -l /dev/rmt2h UDTUUCP400
```

4.3 Adding Subsets

To add a subset to your system, use the `setld` command with the `-l` (load) option. The format is as follows:

`setld` [**–D** *dir*] `-l` <*location*> *subset* [*subset*...]

For example, to reinstall the UDTUUCP400 subset on your system with the software currently residing at `/dev/rmt1h` and you want to add it to your directory at `/mnt`. To do this, type:

```
# setld /mnt /dev/rmt1h UDTUUCP400
```

4.4 Deleting Subsets

To delete a subset from your system, use the `setld` command with the `-d` option. The syntax is as follows:

setld [**–D** *dir*] `-d` *subset* [*subset*...]

For example, to delete the UDTUUCP400 subset from your system, type the following:

```
# setld  -d UDTUUCP400
```

Monitoring and Managing System Performance 5

This chapter provides guidelines that you can use to monitor and to manage system performance. The chapter points out the system-level reports that you can produce to assist you in monitoring the system. It also offers guidelines for particular system management and optimization tasks.

Specifically, the chapter discusses the following:

- Managing system scheduling priority
- Generating system accounting information
- Checking Interprocess Communications facilities status

5.1 Managing Process Scheduling Priority

You can manage the system's process scheduling priority of a given process using the `renice` command. The following example shows how to lower the priority of a process:

```
# /etc/renice +5 -u name
```

Replace `name` with the login ID of the owner of the process.

As the superuser, you can raise the scheduling priority of a user's processes by using a negative number instead of a positive number. For example:

```
# /etc/renice -5 -u name
```

You can raise or lower process scheduling priority on a scale from +20 to –20. For further information, see `renice`(8) in the *ULTRIX Reference Pages*.

5.2 Generating System Accounting Information

There are two types of system accounting information: accumulated and archived. During daily operations, system accounting information is accumulated so that you can keep track of day-to-day operations such as the following:

- User logins
- Command usage
- Printer usage

The following sections describe the commands you can use to display archived system statistics. Note that you must install the optional software subset for accounting to use many of these commands.

5.2.1 Generating User Log-In Report

The system automatically maintains two log-in accounting files: /etc/utmp and /usr/adm/wtmp. The system records all active logins in /etc/utmp and accumulates a user log-in history in /usr/adm/wtmp.

You can generate a report of the system's login-history with the ac command:

```
# /etc/ac -p
```

Over time, /usr/adm/wtmp increases in size. After you generate a hardcopy of the file you should clear it. To clear the /usr/adm/wtmp file, use the cp command with the arguments as follows:

```
# cp /dev/null /usr/adm/wtmp
```

This command copies /dev/null to /usr/adm/wtmp. That is, it reduces /usr/adm/wtmp to a zero-length file.

Note

The system automatically enables log-in history, but it accumulates a log-in history only if /usr/adm/wtmp exists. To disable the system log-in history, remove /usr/adm/wtmp.

For further information, see cp(1) and ac(8).

5.2.2 Generating Command Usage Report

During multiuser startup, /etc/rc normally enables system process accounting. When process accounting is enabled, the system records information on each executed process in /usr/adm/acct. In some systems, system process accounting may be disabled to save disk space.

You can display the contents of the system's current process accounting file, /usr/adm/acct, using the sa command. For example:

```
# /etc/sa
```

This report shows which commands are being used most often on the system.

The file /usr/adm/acct increases in size depending upon your system's activity. To manage space on your /usr file system, you should condense the process accounting information as necessary. To condense /usr/adm/acct, use the sa command with the -s option specified. For example:

```
# /etc/sa -s
```

This command merges the current information in /usr/adm/acct into the process history file, /usr/adm/savacct.

Note

To disable process accounting immediately, type:

```
# /etc/accton
```

To disable process accounting the next time the system reboots, comment out this line in the `/etc/rc` file by putting a # in the first column of the line on which the statement appears. This makes the `accton` line a comment which is not executed. For example:

```
# /etc/accton /usr/adm/acct; echo -n ' accounting' > /dev/console
```

For further information, see `sa`(8) in the *ULTRIX Reference Pages*.

5.2.3 Generating Printer Usage Report

Your system records all printer information in the default accounting file named in `/etc/printcap` if a default accounting file is specified in the `/etc/printcap` file.

To generate a report of your printer usage, use the `pac` command. For example:

```
# /etc/pac
```

The `pac` command displays a report detailing usage per user: number of pages printed, feet of paper consumed, and total estimated cost. For further information, see `printcap`(5) and `pac`(8) in the *ULTRIX Reference Pages*.

5.2.4 Generating Active System Report

In addition to those commands that are used to display accumulated system accounting information, the system has a number of commands that you can use to display active system statistics.

`iostat(1)`	Displays a report of current I/O statistics.
`ps(1)`	Displays a report of the system's process status.
`uptime(1)`	Displays a report of how long the system has been up.
`vmstat(1)`	Displays a report of virtual memory statistics.
`w(1)`	Displays a report of currently active users and what they are doing.
`pstat(8)`	Displays various system tables.
`netstat(1)`	Displays network activity.
`nfsstat(8nfs)`	Displays activity on the Network File System (NFS).

For further information on these commands, see the *ULTRIX Reference Pages*.

5.3 Check Interprocess Communications Facilities Status

Some of the interprocess communications (IPC) facilities requested and used by various processes are not automatically released when the processes exit. You should alert system users to release the facilities after the processes are finished with them. The users can either include an explicit system call in the program to release the facilities, or they can release the facilities using the `ipcrm` command after the process exits. The facilities that must be released are shared memory, semaphores, and message queues.

If the users release the facilities properly, there is no need to administer them. However, if the number of available message queues, shared memory, or semaphores becomes exhausted, then use the `ipcs` command to see the status of these resources. If necessary, you can release them by using the `ipcrm` command. However, make sure that the users are finished with the resources before you release them. You should also show the users how to clean up the resources after having finished using them.

This example shows how to use the `ipcs` command and gives a sample output:

```
# ipcs
IPC status from /dev/kmem as of Fri Jul 19 07:36:20 1985
Message Queues:
T     ID     KEY        MODE       OWNER    GROUP
*** No message queues are currently defined ***

Shared Memory
T     ID     KEY        MODE       OWNER    GROUP
m    400  1627788395  --rw-------    gdp     staff

Semaphores
T     ID     KEY        MODE       OWNER    GROUP
s      2  1644565427  --ra-------    gdp     staff
```

You must be logged in as superuser to release facilities owned by another user. To release the semaphore found in the example, type the following:

```
# ipcrm -s 2
```

The `-s` indicates that a semaphore is to be released, and `2` is the unique ID of the semaphore that was reported in the `ipcs` output. For further information, see `ipcrm`(1) and `ipcs`(1) in the *ULTRIX Reference Pages*.

Device Mnemonics A

This appendix identifies and defines the mnemonics that are used to attach any hardware or software device to your system. The mnemonics are used by the `/dev/MAKEDEV` shell script to create the character or block special files that represent each of the devices. The mnemonics also appear in the system configuration file, as described in the *ULTRIX Guide to Configuration File Maintenance*.

Table A-1 lists the mnemonics in nine categories: generic, systems, consoles, disks, tapes, terminals, modems, printers, and others. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The systems category lists the mnemonic for the DECstation 3100 system setup. The consoles category lists the system console devices that the ULTRIX operating system uses. The disks, tapes, terminals, modems, and printers categories identify the appropriate mnemonics for those devices. The others category lists the mnemonic for DECstation 3100 devices.

The description heading in Table A-1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to both the `MAKEDEV` script and the system configuration file, refer to the reference pages in Section 4 of the *ULTRIX Reference Pages*. If on-line reference pages are available, you can also use the `man` command. For instance, enter the following command at the system prompt to display the reference page for the Mass Storage Control Protocol (MSCP) disk controller driver:

```
% man ra
```

Where appropriate, the SYNTAX section of the reference page defines the device's syntax as it should appear, in the `config` file. Refer to `/dev/MAKEDEV` for additional software device mnemonics that `MAKEDEV` uses. Refer to `MAKEDEV`(8) in the *ULTRIX Reference Pages* for a description of the `MAKEDEV` utility.

Table A-1 uses the convention of an asterisk (*) beside a mnemonic and a question mark (?) beside a device name to mean a variable number. The value of the variable number is dependent on the particular device.

Table A-1: Devices Supported by MAKEDEV

Category	Mnemonic	Description
Generic	boot*	Boot and std devices by cpu number; for example, boot750
	mvax*	All MicroVAX setups; for example, mvax2000
	vaxstation*	A VAXstation 2000 setup; for example, vaxstation2000
	std	Standard devices with all console subsystems
	drum	Kernel drum device
	errlog	Error log device
	audit	Audit log device
	kUmem	Kernel Unibus/Q-bus virtual memory
	kmem	Virtual main memory
	mem	Physical memory
	null	A null device
	trace	A trace device
	tty	A character terminal device
	local	Customer-specific devices
Systems	DECstation	A DECstation 3100 setup
Consoles	console	System console interface
	crl	Console RL02 disk interface for VAX 86?0
	cs*	Console RX50 floppy interface for VAX 8??0
	ctu*	Console TU58 cassette interface for VAX 11/725/730/750
	cty*	Console extra serial line units for VAX 8??0
	cfl	Console RX01 floppy interface for 11/78?
	ttycp	Console line used as auxiliary terminal port
Disks	hp*	MASSBUS disk interface for RM?? drives and RP?? devices
	ra*	UNIBUS/Q-bus/BI/HSC/DSSI MSCP disk controller interface
	rb*	UNIBUS IDC RL02 disk controller interface for RB?? drives
	rd*	VAXstation 2000 and MicroVAX 2000 RD type drives
	rz	SCSI disks (RZ22/RZ23/RZ55/RRD40)
	rk*	UNIBUS RK?? disk controller interface
	rl*	UNIBUS/Q-bus RL?? disk controller interface
	rx*	VAXstation 2000 and MicroVAX 2000 RX type drives
Tapes	mu*	TU78 MASSBUS magtape interface
	tms*	UNIBUS/Q-bus/BI/HSC/DSSI TMSCP tape controller interface
	rv*	UNIBUS/Q-bus/BI TMSCP optical disk
	ts*	UNIBUS/Q-bus TS11/TS05/TU80 magtape interface
	tu*	TE16/TU45/TU77 MASSBUS magtape interface
	st*	VAXstation 2000 and MicroVAX 2000 TZK50 cartridge tape
	tz*	SCSI tapes (TZ30/TZK50)
Terminals	cxa*	Q–bus cxa16
	cxb*	Q–bus cxb16
	cxy*	Q–bus cxt08
	dfa*	Q–bus DFA01 comm multiplexer
	dhq*	Q–bus DHQ11 comm multiplexer
	dhu*	UNIBUS DHU11 comm multiplexer

Table A-1: (continued)

Category	Mnemonic	Description
	dhv*	Q–bus DHV11 comm multiplexer
	dmb*	BI DMB32 comm multiplexer including dmbsp serial printer/plotter
	dhb*	BI DHB32 comm multiplexer
	dmf*	UNIBUS DMF32 comm multiplexer including dmfsp serial printer/plotter
	dmz*	UNIBUS DMZ32 comm multiplexer
	dz	UNIBUS DZ11 and DZ32 comm multiplexer
	sh*	MicroVAX 2000, 8 serial line expansion option
	ss*	VAXstation 2000 and MicroVAX 2000 basic 4 serial line unit
	fc*	VAXstation 60 basic 4 serial line unit
	dzq*	Q–bus DZQ11 comm multiplexer
	dzv*	Q–bus DZV11 comm multiplexer
	lta*	Sets of 16 network local area terminals (LAT)
	pty*	Sets of 16 network pseudoterminals
	qd*	Q–bus VCB02 (QDSS) graphics controller/console
	qv*	Q–bus VCB01 (QVSS) graphics controller/console
	sm*	VAXstation 2000 monochrome bitmap graphics/console
	sg*	VAXstation 2000 color bitmap graphics console
	lx	VAXstation 8000 color high-performance 3D graphics
	fg*	VAXstation 60 color bitmap graphics/console
Modems	dfa*	DFA01 integral modem communications device.
Printers	dmbsp*	BI DMB32 serial printer/plotter
	dmfsp*	UNIBUS DMF32 serial printer/plotter
	lp*	UNIBUS LP11 parallel line printer
	lpv*	Q–bus LP11 parallel line printer
Packet filter	pfilt	Packet filter devices; set of 64
Other	pm*	mono/color bitmap graphics/mouse/modem /printer/terminals for DECstation 3100

Support of the CI/HSC Hardware B

The CI is a high speed, dual-path bus that connects processors and intelligent I/O subsystems (HSCs) in a computer room environment. The HSC is a self-contained, intelligent mass storage controller that provides access to disks and tapes from multiple host nodes attached to the CI bus.

Note

The ULTRIX implementation has the following limitations:

- A maximum of four HSC controllers may be attached to a CI bus.
- A single CI bus may be attached to a host.
- Under no circumstances can an ULTRIX node participate as a VMS cluster member. A configuration which includes a VMS system and an ULTRIX system residing on the same CI is not supported.

This version of ULTRIX supports Digital's System Communication Architecture (SCA) for CI port adapters and HSC controllers. SCA implements port and class driver support, and standardizes the ways in which TMSCP (tms) and MSCP (ra) devices are handled. SCA separates functionality into different architectural layers, thus minimizing the effect that software changes to one layer have on other layers.

B.1 Hardware Setup and Restrictions

For information on physical components and setup, refer to the HSC hardware documentation and the hardware documentation for your processor and supported devices. Only processors with CI adapters can support HSC configurations.

When setting up the HSC hardware, you should attach a terminal to the HSC in order to use commands to get/set HSC parameters, monitor connections between remote systems, and identify the disk/tape status.

The maximum number of hosts on a CI is 16. The host number for any host on the CI must be between 0 and 15; however, if the broadcast address has been set to 0, then 0 cannot be a host number.

Note

Two parameters of particular importance are the system ID and the system name. Use the HSC `SET` command to specify these parameters. Do not duplicate any system identification or names of nodes on the star coupler.

B.2 Software Installation and Restrictions

The installation software assists you in identifying and configuring the components of your system. You should be familiar with the Basic Installation Guide for your processor before starting the actual installation.

During installation of the ULTRIX software, each accessible MSCP (ra) disk device must be uniquely identified by its unit plug number:

- The unit plug number must be between 0 and 254 inclusive.
- Each unit plug number must be unique. Two disks cannot have the same unit plug number even if the disks are on separate controllers. For example, if the unit plug number for a disk on controller A is 5, and the unit plug number for a disk on controller B is also 5, you must change one of the numbers.

After installation, the unit plug numbers can be between 0 and 254 inclusive, and they need not be unique in cases where the disks are on separate controllers.

The CI network device (`scs0`) is not configured by default. The network setup installation script gives you the option to install or not.

B.2.1 Hardware Revision Levels

The correct operation of the software subsystems is sensitive to the revision levels of the CI/HSC microcode. In particular, the following microcode levels should be installed if they are not already:

- HSC microcode level should be V3.9A or higher.
- HSC microcode level V500 or later is needed to support the TA90E and to use the exclusive access functionality provided by the `-e` and `-n` options of the `radisk`(8) utility. See the *ULTRIX Reference Pages* for more information on this utility.
- HSC tape interface boards should have microcode at level 26 or higher.
- HSC disk interface boards should have microcode at level 39 or higher.

Note

This version of ULTRIX, does not support the new CI CISCE 24-node upgrade. The CI microcode distributed by ULTRIX does not support rev. 20 link modules. As a result, the system will be unable to load and verify the CI functional microcode.

B.3 Configuration File Entries

The installation software ensures that your HSC components are configured into the kernel and included in the system configuration file, `/usr/sys/mips/conf/`*HOSTNAME*, where *HOSTNAME* is your system's name, in uppercase letters.

The *ULTRIX Guide to Configuration File Maintenance* provides information on the various entries that correspond to a CI/HSC configuration:

- Description of the `scs_sysid` parameter

- CI adapter specifications
- Controller and device specifications
- The `scsnet` pseudodevice definition

B.4 Booting an HSC Controller or an HSC Disk

If an HSC controller fails, any disks connected to that HSC become inaccessible. Attempts to access those disks will cause the accessing system to hang until the HSC reboots completely.

The ULTRIX software supports booting an HSC disk on all VAX processors with the exception of the MicroVAX class of system. The *ULTRIX Guide to Shutdown and Startup* provides explicit instructions for booting an HSC disk on each processor that supports an HSC configuration.

B.5 Sharing Disk/Tape Units Among Several Hosts

Although an HSC can be shared among several hosts, there is no software interlocking mechanism to prevent concurrent writes to the same partition by multiple ULTRIX systems. The following restrictions must be observed:

- A disk unit can be shared by multiple readers only. Writeable file systems cannot be shared.
- If a disk will be shared, it should be hardware write-protected.
- Each host must mount the file systems to be accessed with the read-only (`-r`) option of the `mount` command.
- Only a single host may mount a disk containing writeable file systems.
- Use the Network File System (NFS) if multiple writers need to share partitions.

You should coordinate disk unit ownership among the hosts on the CI. For example, assign a range of disk unit numbers to each host. The HSC can also be directed to limit disk access to an exclusive host system. This protects the disk from accidental access by another host on the CI. For more information, see the `-e` and `-n` options for the `radisk`(8) utility in the *ULTRIX Reference Pages*.

Tape drives that are attached to an HSC can be shared. This feature is recommended and provides greater use of tape drives. Be aware that the access mechanism provides serial sharing of the drives, not simultaneous access.

B.6 CI Network Capabilities

ULTRIX also provides host to host communications over the CI through a network driver (`scsnet`). The driver can be accessed through the socket system interface in the same manner as the Ethernet drivers. For a description of the network interface, see `scs`(4) in the *ULTRIX Reference Pages*.

Currently the TCP/UDP/IP protocols are supported. The `scsnet` driver takes full advantage of the block mode capabilities of the CI and is therefore a good means for offloading NFS traffic from the Ethernet.

Figure B-1 illustrates how the CI can be used to share disks via NFS.

Figure B-1: Typical CI Configuration

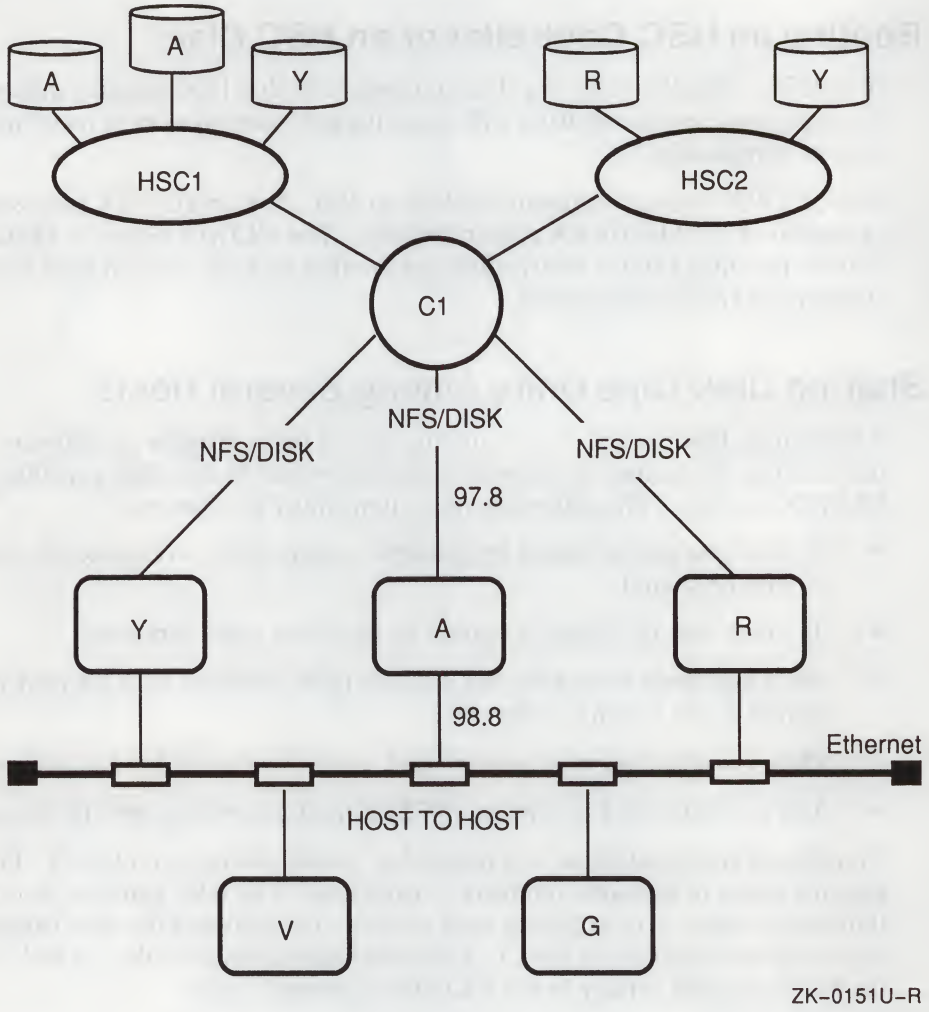

Figure B-1 displays five systems: hosts A, G, R, V, and Y. Hosts Y, A, and R connect to both the CI (net number 97.8) and the Ethernet (net number 98.8). Hosts V and G are connected to the Ethernet only. Note that a separate subset is required for the CI.

Each host shares disks on an HSC. Host A has mounted all of the partitions on the two disks labeled A. Because of the restrictions mentioned above no other system can directly mount A's disks for write access. The only way to obtain access to A's disks is through the NFS. For example hosts Y, and R, could NFS mount A's partitions specifying that the CI path be used instead of the Ethernet. The path is chosen by the `mount`(8) command. For example, the `/etc/hosts` file would have an entry for both paths to system A:

```
"A"      98.8     (for the Ethernet path)
"A-ci"   97.8     (for the CI path)
```

The system manager at system Y would NFS mount a disk partition by specifying host A-ci in the `mount` command instead of host A. For example, the following

command would NFS mount the /usr/users directory from system A using the CI instead of the Ethernet:

```
# /etc/mount A-ci:/usr/users /mnt
```

The NFS traffic would now be routed over the CI and the other host to host traffic over the Ethernet. (Systems V and G would have to specify the Ethernet path because they are not connected to the CI).

A CI must be configured to all systems. The CI is normally configured at boot time; however, the entry for the CI in the /etc/rc.local file must be uncommented and modified to specify the correct broadcast address.

Index

R

S

T

U

V

W

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-343-4040 before placing your electronic, telephone, or direct mail order.

Electronic Orders

To place an order at the Electronic Store, dial 800-234-1998 using a 1200- or 2400-baud modem from anywhere in the USA, Canada, or Puerto Rico. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Your Location	Call	Contact
Continental USA, Alaska, or Hawaii	800-DIGITAL	Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061
Puerto Rico	809-754-7575	Local Digital Subsidiary
Canada	800-267-6215	Digital Equipment of Canada Attn: DECdirect Operations KAO2/2 P.O. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6
International	————	Local Digital subsidiary or approved distributor
Internal*	————	SSB Order Processing - WMO/E15 *or* Software Supply Business Digital Equipment Corporation Westminster, Massachusetts 01473

* For internal orders, you must submit an Internal Software Order Form (EN-01740-07).

Reader's Comments

ULTRIX
Guide to System Environment Setup
AA-ME89B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

Reader's Comments

ULTRIX
Guide to System Environment Setup
AA-ME89B-TE

Please use this postage-paid form to comment on this manual. If you require a written reply to a software problem and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Thank you for your assistance.

Please rate this manual:	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Completeness (enough information)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Page layout (easy to find information)	☐	☐	☐	☐

What would you like to see more/less of? _____

_____

_____

What do you like best about this manual? _____

_____

_____

What do you like least about this manual? _____

_____

_____

Please list errors you have found in this manual:

Page	Description
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Additional comments or suggestions to improve this manual:

_____

_____

_____

_____

What version of the software described by this manual are you using? _____

Name/Title _____ Dept. _____

Company _____ Date _____

Mailing Address _____

_____ Email _____ Phone _____

Do Not Tear – Fold Here and Tape

digital™

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST–CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OPEN SOFTWARE PUBLICATIONS MANAGER
ZKO3–2/Z04
110 SPIT BROOK ROAD
NASHUA NH 03062–9987

Do Not Tear – Fold Here

Cut
Along
Dotted
Line

digital